



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Desarrollo de un Sistema Integral
para la Administración de Servicio
Social**

TESIS

Que para obtener el título de
Ingeniero en Computación

PRESENTAN

Villagómez Pérez Luis Fernando
De La Luz Morales Berenice



DIRECTOR DE TESIS:
Dr. Daniel Trejo Medina

Ciudad Universitaria, Cd. Mx., 2016



UNIVERSIDAD NACIONAL
AVENIDA DE
MEXICO

FACULTAD DE INGENIERÍA
DIVISIÓN DE INGENIERÍA ELÉCTRICA
COMITÉ DE TITULACIÓN

Designación de sinodales de Examen Profesional

A los señores profesores:

Presidente: **MTRO. JUAN JOSE CARREON GRANADOS**
Vocal: **DR. DANIEL TREJO MEDINA**
Secretario: **M.C. ALEJANDRO VELAZQUEZ MENA**
1er. Suplente: **MTRA. EDITH TAPIA RANGEL**
2do. Suplente: **ING. ARIADNA DEL CARMEN TAPIA MIRANDA**

Conforme a la encomienda que hace el Director de la Facultad a este Comité de Titulación para la integración de jurados, me permito informar a ustedes que han sido designados sinodales del Examen Profesional de los señores:

NO. CUENTA	NOMBRE	CARRERA
103005241	VILLAGOMEZ PEREZ LUIS FERNANDO	INGENIERÍA EN COMPUTACIÓN
306156544	DE LA LUZ MORALES BERENICE	INGENIERÍA EN COMPUTACIÓN

quienes han concluido el desarrollo del tema que les fue autorizado.

Ruego a ustedes se sirvan revisar el trabajo adjunto y manifestar a la Dirección de la Facultad, si es el caso, la aceptación del mismo mediante la firma en el oficio FEX-2 en el plazo indicado a continuación.

Por indicaciones del Sr. Director, con el fin de asegurar el pronto cumplimiento de las disposiciones normativas correspondientes y de no afectar innecesariamente los tiempos de titulación, les ruego tomar en consideración que para lo anterior cuentan ustedes con un plazo máximo de **cinco días hábiles** contados a partir del momento en que ustedes **acusen recibo de esta notificación**. Si transcurrido este plazo el interesado no tuviera observaciones de su parte, se entendería que el trabajo ha sido aprobado, por lo que deberán **firmar el oficio de aceptación del trabajo escrito**.

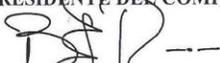
Doy a ustedes las más cumplidas gracias por su atención y les reitero las seguridades de mi consideración más distinguida.

Atentamente,

"POR MI RAZA HABLARÁ EL ESPÍRITU"

Cd. Universitaria, Cd. Mx. a 14 de Septiembre de 2016.

EL PRESIDENTE DEL COMITÉ


DR. BORIS ESCALANTE RAMIREZ

FEX-1
AGV

RESUMEN

Desde su implementación y a través de los años, el Servicio Social Universitario se ha transformado implementando nuevas políticas que satisfacen las necesidades de los estudiantes y de la Universidad, lo que ha llevado a utilizar tecnologías de la información como herramientas de apoyo para la optimización de procesos y mejora en la calidad de los mismos. Sin embargo hemos detectado que la falta de buenas prácticas en la administración de los proyectos de tecnología trajo como consecuencia sistemas de información que nunca fueron actualizados y quedaron obsoletos administrativamente y tecnológicamente hablando.

Por lo anterior se ha enfocado esta tesis al desarrollo de un proyecto de software que, basado en estándares, buenas prácticas y metodologías ágiles, permita a la Universidad contar con un sistema web capaz de incluir, optimizar y actualizar los procesos y lineamientos vigentes del Servicio Social Universitario.

Se busca resolver el problema detectado ofreciendo un proyecto que implemente la metodología de Scrum, abierto a cambios que puedan ser implementados en poco tiempo pero bien documentado de acuerdo a los estándares del IEEE 830 y 1016 correspondientes a la especificación de requerimientos de software y a la descripción del diseño del software respectivamente.

A modo de hipótesis se postula que el desarrollo de este proyecto entregaría valor constantemente a los interesados así como también se considera que la implementación con tecnologías actualizadas para el sistema web, que incorpora las nuevas necesidades administrativas, mejoraría la experiencia de uso, optimizaría los procesos y reduciría el tiempo empleado para el cumplimiento de los mismos.

Después de implementar el sistema web y de concluir el proyecto se ha notado que los resultados sobre el sistema se dieron como habían sido planteados, se cumplieron las hipótesis formuladas obteniendo reducciones significativas en el tiempo de ejecución de algunos procesos a causa de mejoras en el rendimiento del sistema y por la descentralización de algunas de las tareas. Sin embargo durante el desarrollo existieron complicaciones de tipo administrativas que provocaron atrasos significativos de acuerdo a la estimación y planeación original.

En conclusión, se considera este proyecto como exitoso ya que para la Universidad eran prioridad contar con los beneficios derivados de la implementación dejando en segundo plano la metodología de desarrollo utilizada. Por otro lado, como proyecto de Ingeniería de Software, sí es posible hablar de un proyecto que fracasó precisamente por el desfase tan grande que existió entre los tiempos de entrega planeados y los reales.

ABSTRACT

Social Service at the University has been transformed since its implementation through the years, implementing new policies that meet the students and University needs, which has led to use information technologies as tools of support for the optimization of processes and improvement in the quality of them. However we have found that the lack of good practices in the management of technology projects resulted in information systems that were never updated, and became obsolete administrative and technologically speaking.

Therefore we have focused this thesis to the development of a software project based on standards, best practices, and agile methodologies, allowing University to have a web system capable of include, optimize and update their processes and guidelines in force of the University Social Service.

We seek to solve the detected problem offering a project that implements the Scrum methodology, open to changes that can be implemented in a short time but well documented according to the IEEE 830 and 1016 standards for the software requirements specification and the software design description respectively.

As an hypothesis we postulate that the development of this project would deliver constantly value to stakeholders as well as we also consider that the implementation of updated technologies for the web system, that incorporates the new administrative needs, would improve the use experience, optimize processes and would reduce the taken time for the fulfillment of the themselves.

After web system implementation and project conclusion, we could notice that the system results were given as they had been raised, the hypothesis have been approached obtaining significant reductions in the execution time of some processes because of improvements in the performance of the system and also by the decentralization of some tasks. However during the development there were administrative issues that caused significant delays according to the original estimation and planning.

In conclusion, we consider this as a successful project, since for the University were priority the benefits acquired with the implementation leaving in second flat the development methodology used. Instead, as a Software Engineering project, we can talk about a project that failed precisely because the large gap that existed between the actual and planned delivery times.

AGRADECIMIENTOS

Berenice

Quiero agradecer a mis padres que siempre han estado conmigo, y me han acompañado siempre con su amor y consejos. Gracias por apoyarme en todo, por haber estado en mi camino académico hasta este momento de cerrar un ciclo, siempre invitándome a no conformarme, a tener pasión y amor por lo que se hace, no perdiendo el agradecimiento y la humildad. Son un ejemplo para mí y los amo mucho, que Dios les recompense todo lo que han hecho.

También a mis hermanos, que siempre me han dado su ánimo, alegría, también regaños y mucho amor. Siempre han estado conmigo, son mis mejores amigos, con ellos las cosas son mejores.

Gracias a mi amigo Luis por considerarme para colaborar juntos, fue enriquecedor. En verdad gracias Luis.

Igualmente agradezco al profesor Daniel Trejo Medina, profe como le digo, por brindarme la confianza, darme parte de sus conocimientos y tiempo en este trabajo, he aprendido muchas cosas de él y lo considero un ejemplo, mil gracias.

Gracias a mi amada Universidad, por el honor de haber transitado por sus aulas y ser parte de ella, abirme un mundo de conocimiento un sentimiento que embarga mi ser. Gracias a mis profesores, que día tras día compartían sus conocimientos para crear un mundo mejor, gracias a todos los maestros.

Sobre todo agradezco a Dios que me da la oportunidad de vivir y la fortaleza para realizar muchas cosas, porque todo lo puedo en las fuerzas que Él me da y lo que logro es para Él.

A mis padres por darme la vida, por inculcarme buenos valores y principios, por la confianza y apoyo incondicional, por todo el amor que me han brindado y por todo su esfuerzo para que yo tuviera esta educación.

A mi madre por heredarme el carácter y la fuerza para salir adelante ante cualquier adversidad, por su incansable espíritu de lucha que me inspira a ser mejor cada día.

A mi padre por enseñarme con el ejemplo que el compromiso y el esfuerzo siempre serán bien retribuidos, por ser mi guía y motivarme a sobresalir siempre.

A mis hermanos que sin ellos mi vida no sería la misma, por ser mis compañeros y cómplices de vida, por dejarme ser su ejemplo.

A Daniela, mi novia, por llegar para quedarse, por enseñarme el amor, por el infinito apoyo y la eterna motivación, por ser mi inspiración, mi luz y mi fuerza.

A mi familia, a mis incontables primos y primas por quererme tanto y por todos esos días de diversión y convivencia; a mis tíos y tías por ser como un padre o una madre más, a todos por ayudarme a salir adelante en los momentos difíciles.

A mis amigos, en especial a Poncho y Christian por enseñarme que no hay que ser bueno o mejor amigo, simplemente ser, por ser tan incondicionales, por todo su apoyo, por los consejos, pero sobre todo por estar siempre cerca en las malas, en las buenas y en las mejores.

A mis profesores por compartirme sus conocimientos y experiencias, por hacer de mí un ingeniero con ética y digno de la Universidad.

A mis sinodales de tesis por el tiempo dedicado para corregir este trabajo, por el aprendizaje que me dejaron y por su invaluable apoyo.

A mi compañera de tesis y amiga, por aceptar trabajar juntos, por su esfuerzo, compromiso y entrega, pero sobre todo por ayudarme a cumplir esta meta.

A mi profesor, amigo y director de tesis Dr. Daniel Trejo Medina por ser el mejor profesor que la Facultad haya visto en mucho tiempo, por compartir su conocimiento y sabiduría, por enseñarme que los mejores somos de la UNAM, por la oportunidad

de hacerme ingeniero con sus consejos, por los regaños, por los aplausos, por su sinceridad y por su incansable e insistente apoyo, por todo.

A la Dirección General de Orientación y Atención Educativa y a la Dirección de Servicio Social por la oportunidad de trabajar ahí, por la confianza para desarrollar el proyecto y por todas las facilidades que me han brindado durante estos años.

A la Facultad de Ingeniería por abrirme sus puertas, por prestar sus aulas para mi formación y por las lecciones que me dejó pisar tan emblemático sitio.

A la Universidad Nacional Autónoma de México por adoptarme desde que era un niño, por haberme hecho todo lo que soy y haberme enseñado todo lo que sé, por permitirme formar parte de una selecta comunidad del pensamiento, por contribuir en la formación de mejores seres humanos para esta nación, por enseñarme el compromiso que ahora tengo con la sociedad, por permitirme expresar libremente sin juzgar y por formarme como ingeniero de la máxima casa de estudios.

A todos y cada uno de ustedes que hicieron esto posible,
mi eterno agradecimiento.

POR MI RAZA HABLARÁ EL ESPÍRITU
Luis F.

ÍNDICE

RESUMEN	I
ABSTRACT	III
AGRADECIMIENTOS	V
ÍNDICE	XI
ÍNDICE DE ILUSTRACIONES	XIII
ÍNDICE DE TABLAS	XIV
INTRODUCCIÓN	1
ANTECEDENTES	5
EL SERVICIO SOCIAL UNIVERSITARIO (SSU)	5
<i>Definición</i>	5
<i>Misión, visión y objetivos</i>	6
<i>Marco legal</i>	7
<i>Estructura</i>	7
POLÍTICAS Y PROCEDIMIENTOS DEL SSU	8
<i>Instituciones receptoras</i>	8
<i>La Dirección General de Orientación y Servicios Educativos (DGOSE)</i>	9
<i>Prestadores de servicio social</i>	9
<i>Procesos</i>	9
SISTEMA DE INFORMACIÓN AUTOMATIZADA DE SERVICIO SOCIAL (SIASS)	12
PROBLEMA A RESOLVER	13
TIPO DE ESTUDIO (PROPUESTA METODOLÓGICA)	14
OBJETIVO GENERAL	15
OBJETIVOS PARTICULARES	15
JUSTIFICACIÓN	16
HIPÓTESIS	18
MARCO DE REFERENCIA	19
INGENIERÍA DE SOFTWARE	23
MODELOS DEL PROCESO DE SOFTWARE	23
<i>Cascada</i>	23
<i>Evolutivo</i>	25
<i>Componentes</i>	26
<i>Proceso Unificado de Rational (RUP)</i>	26
PROCESO DEL SOFTWARE	28
<i>Requerimientos del Software</i>	29
<i>Diseño del Software</i>	31
ADMINISTRACIÓN DE PROYECTOS	37
<i>Riesgos y Control de Riesgos</i>	38
<i>Planificación de Proyectos de Software</i>	39
METODOLOGÍAS ÁGILES	40
<i>Manifiesto Ágil</i>	41

<i>Scrum</i>	41
EVALUACIÓN Y MEJORA DEL PROCESO	48
<i>Modelo Integración del Modelo de Madurez de Capacidades (CMMI)</i>	49
<i>Modelo de Madurez de Capacidades del Personal (People-CMM)</i>	51
BASES DE DATOS RELACIONALES	53
ARQUITECTURA MODELO, VISTA, CONTROLADOR (MVC)	54
MANEJADOR DE VERSIONES SUBVERSION (SVN)	55
DESARROLLO	57
INVESTIGACIÓN DE LOS ANTECEDENTES.....	57
DEFINICIÓN Y FACTIBILIDAD DEL PROYECTO	57
INICIO DEL PROYECTO	59
REQUERIMIENTOS GENERALES.....	61
ACUERDO DE TRABAJO PARA EL DESARROLLO (STATEMENT OF WORK, SOW).....	61
ESTIMACIÓN INICIAL DEL TIEMPO DE DESARROLLO.....	62
COSTO ESTIMADO INICIAL DEL PROYECTO	66
DISEÑO DE LA ARQUITECTURA DEL SOFTWARE	67
IMPLEMENTACIÓN DE LA BASE DE DATOS.....	67
DESARROLLO DEL SISTEMA.....	69
PRUEBAS FUNCIONALES Y DE SEGURIDAD	69
RESULTADOS	70
CONCLUSIONES	83
REFERENCIAS	89
ANEXOS	93
ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (<i>SOFTWARE REQUIREMENTS SPECIFICATION</i>)	93
DESCRIPCIÓN DEL DISEÑO DEL SOFTWARE (<i>SOFTWARE DESIGN DESCRIPTION</i>)	109
MANUAL DE USUARIO – INSTITUCIONES RECEPTORAS	127

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Proceso General del Servicio Social Universitario.	10
Ilustración 2 El ciclo de vida del Software.	24
Ilustración 3 Desarrollo evolutivo.....	25
Ilustración 4 Modelo del diseño.	33
Ilustración 5. Flujo del Scrum.	46
Ilustración 6 Áreas de proceso en las representaciones continua y por etapas.	50
Ilustración 7 Los cinco niveles de madurez del P-CMM.	52
Ilustración 8 Funcionamiento de la arquitectura MVC.	55
Ilustración 9 Un sistema típico cliente/servidor.....	56
Ilustración 10 Análisis FODA del proyecto.	59
Ilustración 11 Gráfica Gantt inicial del proyecto.	62
Ilustración 12 Gráfica Gantt inicial del proyecto.	63
Ilustración 13 Gráfica Gantt inicial del análisis del proyecto.....	66
Ilustración 14 Diagrama de diseño de la base de datos.	68
Ilustración 15 Consulta de programas – Inicio.....	71
Ilustración 16 Consulta de programas – Filtro de búsqueda avanzada.....	71
Ilustración 17 Consulta de programas – Listado de resultados.	72
Ilustración 18 Consulta de programas – Detalle del programa.....	72
Ilustración 19 SIASS – Inicio.	74
Ilustración 20 Registro de Usuario IR (1).	74
Ilustración 21 Registro de Usuario IR (2).	75
Ilustración 22 Registro de Usuario IR (3).	75
Ilustración 23 Registro de programa de servicio social (1).....	76
Ilustración 24 Registro de programa de servicio social (2).	76
Ilustración 25 Registro de programa de servicio social (3).	77
Ilustración 26 Registro de programa de servicio social (4).	77
Ilustración 27 Registro de programa de servicio social (5).	77
Ilustración 28 Evaluación de programas – Filtro de búsqueda.	78
Ilustración 29 Evaluación de programas – Listado de programas.	79
Ilustración 30 Registro de prestadores – Buscar alumno.	80
Ilustración 31 Generación de reportes – Programas Anuales.	81
Ilustración 32 Generación de reportes – Programas por eje de acción.....	81
Ilustración 33 Desviación de entrega de proyecto.....	82
Ilustración 34 Flujo de operación.....	101
Ilustración 35 Funcionalidades del SIASS.....	112
Ilustración 36 Módulos del SIASS.	112
Ilustración 37 Gestión institucional.	113
Ilustración 38 Administración de programas de servicio social.	113
Ilustración 39 Gestión escolar.	114
Ilustración 40 Consulta de programas.....	114
Ilustración 41 Generador de reportes.	115
Ilustración 42 Certificación electrónica.	115
Ilustración 43 Diagrama de flujo del SIASS.....	116
Ilustración 44 Arquitectura cliente servidor del SIASS.	117

Ilustración 45 MVC del SIASS.....	117
Ilustración 46 Diagrama de componentes del SIASS con DGAE.....	118
Ilustración 47 Diagrama entidad relación del SIASS.....	119
Ilustración 48 Consulta de programas (1).	120
Ilustración 49 Consulta de programas (2).	120
Ilustración 50 Consulta de programas (3).	121
Ilustración 51 Consulta de programas (4).	122
Ilustración 52 Registro de prestadores.....	123
Ilustración 53 Registro de usuarios DGOSE.	124
Ilustración 54 Registro de usuarios URSS.	125
Ilustración 55 Registro de usuarios IR.....	126

ÍNDICE DE TABLAS

Tabla 1 Resumen de viewpoint.	36
Tabla 2 Estimación de Sprints.....	65
Tabla 3 Sprint de planeación y análisis.	65
Tabla 4 Estimados de costo.	82
Tabla 5 Lenguajes y tecnologías del SIASS.	110
Tabla 6 Resumen de design concerns correspondientes al SIASS.	111

INTRODUCCIÓN

La Universidad Nacional Autónoma de México es la Institución educativa de mayor prestigio en el país. A nivel licenciatura, cada año transitan por sus aulas más de doscientos mil alumnos de 117 carreras, sin embargo el porcentaje de titulación anual apenas rebasa lo diez puntos en promedio. La principal causa de estos números tan bajos sin duda es la deserción escolar por cualquiera que sea el motivo; sin embargo existen otras razones por las que los alumnos y sobre todo los pasantes no logran conseguir el grado, entre esas razones creemos que se encuentran la dificultad para superar el requisito de idioma que la mayoría de las carreras solicita y el requisito de servicio social, que por reglamento es obligatorio para todas las carreras de licenciatura.

Desafortunadamente en la Universidad, muchos estudiantes de licenciatura ven al servicio social solo como un trámite o requisito para lograr la titulación. Y si a eso le agregamos el hecho de que se debe cumplir una serie de requisitos y trámites para poder realizar el servicio social, entonces los estudiantes con mayor facilidad se pueden inclinar por una postura de desagrado hacia este tema.

Lo cierto es que, más allá de ser un requisito de titulación lleno de trámites, el Servicio Social Universitario representa una excelente oportunidad para los estudiantes y pasantes de demostrar e implementar los conocimientos adquiridos a lo largo de su carrera, de adquirir experiencia profesional en la resolución de problemas y sobre todo de retribuir, a través del conocimiento, un poco de todo lo que la sociedad nos ha brindado.

Para enfrentar el problema mencionado y retomar el verdadero sentido formativo, social y retributivo del servicio social, la Universidad ha planteado nuevas estrategias y lineamientos que buscan impulsar el desarrollo humano, ético y profesional de sus egresados, de tal manera que adquieran un enfoque social y solidario para intervenir en la solución de problemas y necesidades básicas de nuestro país.

Realizar el servicio social implica que el alumno deba completar una serie de requisitos y procesos, en donde se ven involucradas diversas dependencias de la Universidad e instituciones del sector público y social que reciben a los prestadores. Por naturaleza, la participación de diferentes entidades en un mismo proceso, provoca trámites burocráticos y administrativos que pueden llegar a ser complicados y hasta tediosos para los alumnos acostumbrados a la vida más práctica y ágil.

Una de las estrategias tomadas por la Universidad para atender este problema ha sido el uso y la implementación de nuevas tecnologías que acerquen a los alumnos a lo que es el Servicio Social Universitario sin que tengan que desplazarse a lugares específicos, agilizando la ejecución de sus trámites y liberándolos de esa carga que

mayormente recaía sobre ellos, así mismo ofreciendo un panorama de opciones diversas para que se sientan motivados a realizar esta actividad.

Es así como nace la idea original que da pauta al desarrollo de ésta tesis, llevar a cabo la implementación de un nuevo sistema web para la administración del Servicio Social Universitario, mismo sistema que retomaría el nombre de su antecesor: Sistema de Información Automatizada de Servicio Social (SIASS).

El sistema facilitará y optimizará los procesos que se llevan a cabo durante el trámite del Servicio Social Universitario. En este documento se presenta: la problemática a fondo, la propuesta de mejora, el marco de conocimientos que sirvió de referencia y las tareas que se realizaron para el cometido.

A lo largo de esta tesis, iremos presentando las áreas de oportunidad que se detectaron en el Servicio Social Universitario y que fueron incluidas en el desarrollo del sistema, mismas que al resolverse consideramos resultarán benéficas para todos los usuarios, así como aquellas oportunidades que se presentaron para innovar y dar propuestas contemporáneas que en primera instancia satisfacen a las necesidades de la Universidad.

Por sentido común sabemos que para resolver un problema, primero es necesario contar con uno. Bajo esta afirmación y conociendo los antecedentes nos dimos a la tarea de estructurarlo de tal forma que englobara todos los antecedentes y dificultades que se presentaban y así formular en conjunto una solución.

Con el problema planteado, procedimos a fijar un objetivo para el proyecto, mismo que buscaría cumplirse a través de la ejecución de la solución formulada. Una vez que teníamos claros y definidos un problema, un objetivo y una solución establecimos un par de hipótesis que de acuerdo a la experiencia creemos que se cumplirán al finalizar la metodología.

Durante este análisis encontramos puntos de debilidad en la administración del software de la Dirección General de Orientación y Servicios Educativos, lo cual fue una de las causas principales del problema. Por tal motivo, la solución planteada en su mayoría se orienta a mejorar la manera en la que se administran los proyectos y el software de la dependencia, sobre todo en materia de servicio social.

Como parte de la solución al problema y atendiendo a las necesidades de innovación decidimos realizar un proyecto de software utilizando una metodología de desarrollo ágil llamada Scrum, que actualmente está en boga por el beneficio que aporta al ciclo de vida del software y sobre todo por el valor directo e inmediato que ofrece al cliente. La razón por la que consideramos Scrum como la mejor opción se debe principalmente a la facilidad de incluir cambios en los requerimientos de software y porque ofrece entregas funcionales en periodos cortos de tiempo, situaciones que por experiencia sabemos se solicitan constantemente en la Universidad.

Con la puesta en marcha del proyecto, detectamos que algunas de las áreas de oportunidad para mejorar eran más complejas de lo estimado, pues las circunstancias no siempre fueron favorables para la resolución de los problemas, ya que nos enfrentamos con que el cumplimiento de nuestras propuestas no dependía totalmente de la administración y desarrollo del proyecto, sino también de factores externos al equipo de trabajo como son los demás involucrados en el proyecto y la resolución del problema mismo.

A pesar de los contratiempos e inconvenientes surgidos durante el proyecto, al finalizar el sistema pudimos afirmar que se cumplieron los objetivos planteados, que las hipótesis fueron verdaderas y lo más importante que el resultado realmente beneficia a todos los involucrados en las diversas actividades del Servicio Social Universitario.

Para finalizar con esta breve introducción esperamos que esta tesis sirva como modelo para plantear la administración de un proyecto de software utilizando metodologías ágiles y promoviendo el uso de normas, estándares y herramientas que la Ingeniería de Software sugiere. Así mismo buscamos promover la estructura adecuada que una tesis de licenciatura debe contener, resolviendo un problema a través del uso del método científico, incluyendo planteamiento del problema, objetivos, hipótesis, resultados y conclusiones basadas en las hipótesis formuladas.

Por último hacemos una atenta invitación a todos los profesores de la Universidad para que motiven a sus alumnos a titularse a través de la modalidad de tesis o tesina y examen profesional ya que amplía el panorama de los estudiantes y permite desarrollar habilidades que otras opciones de titulación no ofrecen.

ANTECEDENTES

El Servicio Social Universitario (SSU)

Definición

De acuerdo con el Reglamento General de Servicio Social (RGSS) de la Universidad Nacional Autónoma de México (1985): “se entiende por servicio social universitario la realización obligatoria de actividades temporales que ejecuten los estudiantes de carreras técnicas y profesionales, tendientes a la aplicación de los conocimientos que hayan obtenido y que impliquen el ejercicio de la práctica profesional en beneficio o en interés de la sociedad.” (UNAM, 1985).

El Servicio Social Universitario (SSU) es parte del proceso de formación de los estudiantes universitarios, además permite que ellos se involucren directamente en el desarrollo de soluciones para atender las problemáticas nacionales que se presentan en los diferentes ámbitos de nuestra sociedad, por lo anterior, la Dirección de Servicio Social de la UNAM (DGOSE, UNAM. 2015) define a esta actividad de la siguiente forma:

El servicio social es una actividad temporal-obligatoria que permite al estudiante:

En el ámbito formativo (entendido como el proceso de construcción de saberes científicos, sociales, artísticos y humanísticos que facilitan el desarrollo de la estructura ética/moral de la persona y dan sentido a la integración del perfil profesional):

- Consolidar la formación académica.
- Poner en práctica los conocimientos adquiridos en las aulas.
- Adquirir nuevos conocimientos y habilidades profesionales.
- Aprender a actuar con solidaridad, reciprocidad y a trabajar en equipo.
- Conocer y tener la posibilidad de incorporarse al mercado de trabajo.

En el ámbito social (participando en los diferentes campos de la actividad humana como son salud, educación, cultura, equidad de género, medio ambiente, producción de bienes y servicios, derechos humanos, ciencia y tecnología; incentivando la fructificación de talentos y capacidades de creación, en la atención/gestión de sus condiciones y problemas, que serán asumidas con autorresponsabilidad):

- Tomar conciencia de la problemática nacional, en particular la de los sectores más desprotegidos del país.
- Extender a la sociedad los beneficios de la ciencia, la técnica y la cultura.

- Poner al servicio de la comunidad los conocimientos, habilidades y destrezas adquiridos.

En el ámbito retributivo (contribuir a la mejora de la calidad de vida, a través de la participación en el diseño, intervención, planificación, asesoría, capacitación, etc., al otorgar a la comunidad los beneficios de las competencias adquiridas, saberes, habilidades y actitudes):

- Reconocer el costo de la educación recibida y retribuir a la sociedad con acciones responsables.
- Privilegiar el compromiso adquirido con la sociedad, atendiendo las necesidades y problemas de los grupos vulnerables.
- Retribuir a la sociedad los recursos destinados a la educación pública.
- Poner en práctica las competencias adquiridas para realizar trabajo comunitario.

Misión, visión y objetivos

Como lo indica en su misión, el servicio social universitario busca en los estudiantes:

“Contribuir a la formación integral del estudiante universitario, mediante la participación y puesta en práctica de sus capacidades en la solución de los problemas y necesidades del país, a través de actividades que fortalezcan su conciencia ética, moral, humanista y de retribución a la sociedad, a la vez que desarrolle habilidades y estrategias que faciliten su incorporación al mercado laboral.” (DGOSE, UNAM. 2015).

Lo anterior está orientado hacia el cumplimiento de la visión que detalla la Universidad a través del portal de servicio social donde se dice que:

“El Servicio Social será una oportunidad de cooperación incluyente y de reciprocidad; un espacio de aprendizaje y fuente del saber; un medio de cultura y educación disponible para todos, que permite el reconocimiento y fortalecimiento de conocimientos, habilidades, destrezas y actitudes; un lugar en donde el arte y la ciencia sean amalgama de la calidad humana y profesional. El servicio social será un tiempo y lugar, en donde impere y se incentive el desarrollo de la excelencia humana.” (DGOSE, UNAM. 2015).

De esta manera se construye el objetivo general, en el cual el servicio social, de acuerdo a lo establecido por la misma Universidad, pretende:

“Impulsar el desarrollo profesional y humano de los estudiantes universitarios, a través del compromiso social activo y solidario en la solución de problemas y/o necesidades del país, esto mediante actitudes cívicas, críticas y positivas en

favor de la consolidación de cuadros de excelencia humana, como base para un futuro próspero de la nación.” (DGOSE, UNAM. 2015).

Complementando el objetivo del SSU, la UNAM a través del RGSS (1985) indica que el servicio social debe cumplir los siguientes puntos:

- I. Extender los beneficios de la ciencia, la técnica y la cultura a la sociedad;
- II. Consolidar la formación académica y capacitación profesional del prestador del servicio social;
- III. Fomentar en el prestador una conciencia de solidaridad con la comunidad a la que pertenece.

Marco legal

De acuerdo con la DGOSE (2015) el Servicio Social Universitario se encuentra regulado por la Ley Reglamentaria del Artículo 5º Constitucional en su capítulo VII.

En la Universidad está normado por:

- El Reglamento General del Servicio Social
- Reglamento General de Estudios Universitarios
- El Reglamento General de Exámenes

Además por los reglamentos de Servicio Social de Facultades, Escuelas, Institutos y Centros autorizados por los Consejos Técnicos y Consejos Académicos:

- Centro de Nanociencias y Nanotecnología
- Escuela Nacional de Trabajo Social
- Facultad de Ciencias
- Facultad de Ingeniería
- Facultad de Medicina Veterinaria y Zootecnia
- Facultad de Música
- Facultad de Química
- Facultad de Estudios Superiores Cuautitlán
- Facultad de Estudios Superiores Zaragoza
- Psicología
- Química Farmacéutico Biológica
- Instituto de Energías Renovables

Estructura

El Servicio Social Universitario está regulado por tres instancias y cada una cumple diferentes funciones específicas dentro del proceso general del servicio social. Estas instancias son:

1. Consejos Técnicos de las facultades y escuelas que:

- a) Establecen las modalidades para el cumplimiento del servicio social en cada una de las facultades y escuelas a través de los respectivos reglamentos internos.
 - b) Proponen programas interdisciplinarios y multidisciplinarios de servicio social.
2. Unidades Responsables de Servicio Social (URSS) de facultades y escuelas encargadas de:
- a) Aprobar, promover, supervisar y evaluar la realización de programas de servicio social.
 - b) Registrar el inicio y término de la prestación del servicio social de los alumnos.
3. Dirección General de Orientación y Servicios Educativos (DGOSE), a través de la Subdirección de Servicio Social, encargada de:
- a) Regular, coordinar, supervisar y evaluar la prestación del servicio social universitario.
 - b) Emitir la Carta Única de Liberación.

Políticas y procedimientos del SSU

Instituciones receptoras

Las instituciones receptoras (IR) son aquellas instituciones del sector público y social que, de acuerdo a los lineamientos vigentes, hayan sido aprobadas por la DGOSE para registrar programas de servicio social que deriven de alguno de los ejes de acción definidos por la Universidad y que además pueden solicitar prestadores de servicio social.

Entre las funciones de las IR, para fines del proceso de servicio social, encontramos:

1. Registrar programas de servicio social asociadas a un eje de acción.
2. Realizar el seguimiento del total de programas registrados y su respectiva evaluación por carrera, así como el número de vacantes para cada perfil solicitado.
3. Solicitar prestadores de servicio social para que participen en los programas que han sido aprobados por las escuelas y/o facultades de la Universidad para las carreras correspondientes.
4. Supervisar las actividades de los prestadores de servicio social y notificar a las URSS cuando el alumno haya concluido satisfactoriamente o haya causado baja, según sea el caso.

La Dirección General de Orientación y Servicios Educativos (DGOSE)

La DGOSE es una dependencia universitaria que entre sus funciones tiene la responsabilidad de administrar los procesos del servicio social universitario y supervisar que estos sean ejecutados, por todos los involucrados, de acuerdo a lo establecido en sus lineamientos internos y en los reglamentos de la Universidad.

Como se menciona en la Estructura del SSU del presente documento, la DGOSE también emite la Carta Única de Liberación de Servicio Social que acredita que el prestador ha concluido las actividades establecidas en el programa de servicio social al que se inscribió; a través de este documento de carácter oficial es como se notifica actualmente a la Dirección General de Administración Escolar (DGAE) o a la Dirección General de Incorporación y Revalidación de Estudios (DGIRE) que el alumno ha concluido en su totalidad el servicio social como parte del proceso de titulación.

Prestadores de servicio social

Los prestadores de servicio social son, en términos generales, aquellos estudiantes que cumplen los requisitos mínimos establecidos para iniciar el servicio social y que se inscriben en alguno de los programas registrados en el sistema. Una vez inscritos a un programa, los prestadores tienen la responsabilidad de cumplir con el mínimo de horas y de presentar los informes que estén establecidos en su Escuela o Facultad.

Procesos

El proceso general de servicio social involucra a las siguientes figuras:

- La DGOSE, que supervisa el cumplimiento de los lineamientos.
- La URSS correspondiente a la escuela o facultad y a la carrera que cursa el estudiante en cuestión.
- La IR a donde llegan los prestadores a efectuar las actividades del servicio social.
- Los alumnos de las carreras impartidas en la UNAM o en el sistema incorporado y que se inscriben a un programa de servicio social.

Basado en la experiencia, se puede decir que el servicio social se ejecuta de acuerdo al siguiente diagrama:

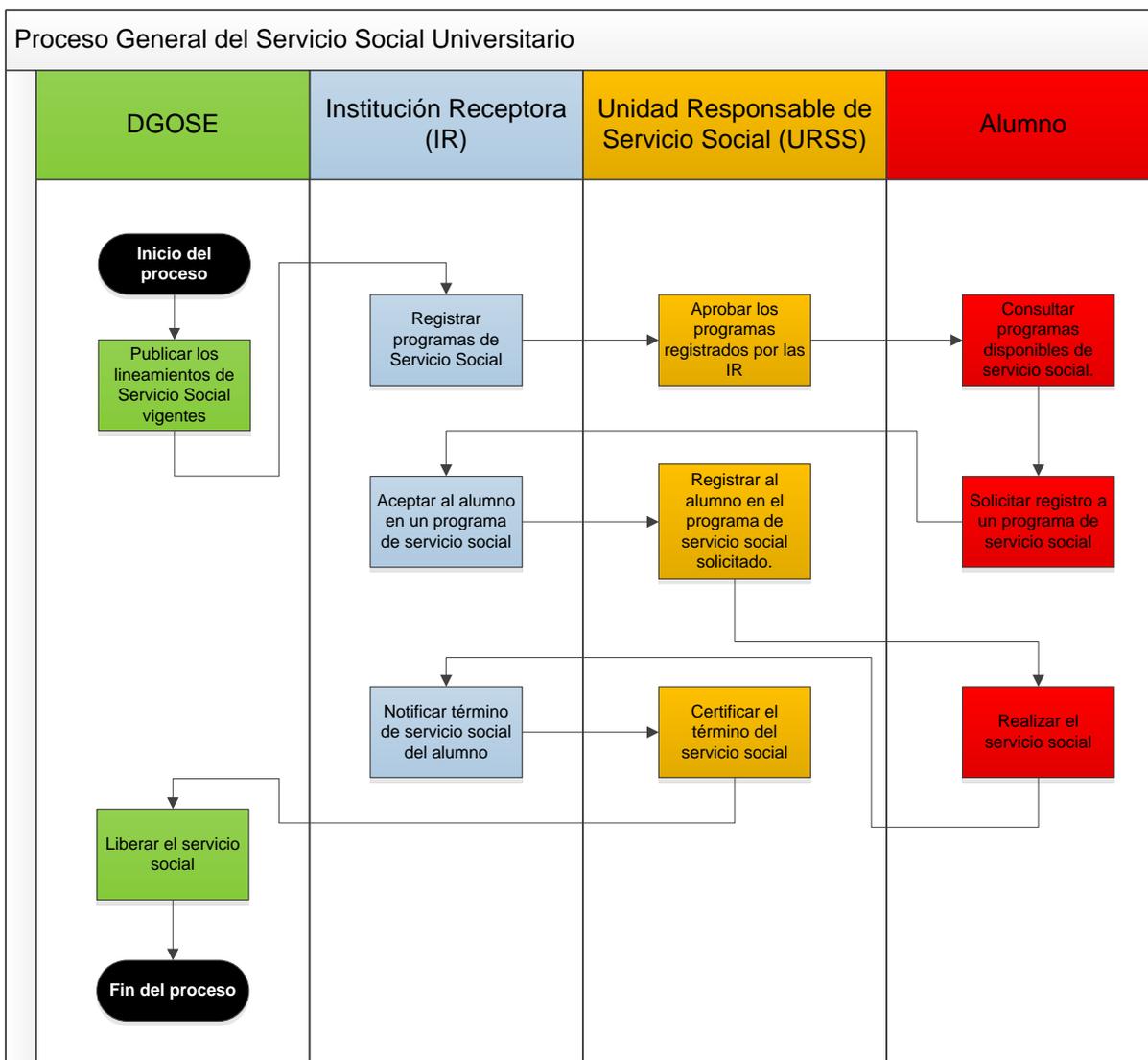


Ilustración 1 Proceso General del Servicio Social Universitario. Fuente: Sitio Web de la DGOSE, adaptación propia

A continuación se describe particularmente cada una de las actividades realizadas dentro de este proceso:

1. **Publicar los lineamientos de Servicio social vigentes.** La DGOSE publicará las nuevas políticas del Servicio Social Universitario para el periodo en curso y los lineamientos para el registro de Instituciones y programas.
2. **Registrar programas de servicio social.** Las IR registrarán los programas de servicio social de acuerdo a sus necesidades y deberán definir las carreras que solicitan y las actividades para cada una de ellas.

3. **Aprobar los programas registrados por la IR.** Las Escuelas y Facultades a través de las URSS evaluarán, individualmente y por carrera, los programas registrados y de acuerdo a su reglamento interno aprobarán aquellos que cumplan y en los cuales sus alumnos podrán participar.
4. **Consultar programas disponibles de servicio social.** Los alumnos podrán revisar todas las diferentes opciones de programas aprobados para su carrera y elegir el que mejor se adapte a sus aptitudes y gustos.
5. **Solicitar el registro a un programa de servicio social.** El alumno que ha elegido un programa deberá presentarse ante la IR para conocer más acerca del programa elegido y sus actividades.
6. **Aceptar a un alumno en un programa de servicio social.** La IR recibe a los alumnos interesados en participar para alguno de los programas y de acuerdo a sus criterios y vacantes, las instituciones emitirán o no la carta de aceptación al alumno.
7. **Registrar al alumno en el programa de servicio social solicitado.** Cuando el alumno ha sido aceptado por la IR, deberá acudir a su escuela o facultad con la documentación correspondiente para solicitar el trámite de registro en el programa a través del sistema de servicio social y así iniciar oficialmente sus actividades como prestador.
8. **Realizar el servicio social.** El prestador de servicio social deberá ejecutar sus actividades indicadas en el programa durante el tiempo establecido por su facultad o por el mismo programa, cumpliendo con la entrega periódica de informes de actividades a la URSS correspondiente.
9. **Notificar término de servicio social del alumno.** Una vez que el alumno haya concluido todas sus actividades dentro del programa de servicio social, la institución deberá notificar a la URSS, mediante un documento, el término de las actividades asignadas al prestador.
10. **Certificar el término de servicio social.** La URSS correspondiente deberá certificar, de acuerdo a sus reglamentos internos, la conclusión de actividades y entrega de documentación por parte del alumno y de esta manera notificar a la DGOSE que dicho alumno ha cumplido satisfactoriamente con el servicio social.
11. **Liberar el servicio social.** La DGOSE como dependencia reguladora del Servicio Social Universitario en la UNAM se encargará de liberar oficialmente esta actividad y notificar, mediante la Carta Única de Liberación, a las instancias correspondientes para los trámites de titulación.

Sistema de Información Automatizada de Servicio Social (SIASS)

En la búsqueda por mejorar y fortalecer el servicio social en la UNAM, la Dirección General de Orientación y Servicios Educativos ha trabajado en diversas acciones de mejora continua. En este sentido y derivado del trabajo colegiado con las Unidades Responsables de Servicio Social, se establecieron diversas normas y lineamientos para el Servicio Social Universitario que permitieron generalizar y establecer procesos que cubrirían la mayoría de las necesidades de las escuelas y facultades.

Fue así que con el apoyo de la entonces Dirección General de Servicios de Cómputo Académico (DGSCA) se desarrolló y se puso en marcha, en noviembre de 2007, el Sistema de Información Automatizada de Servicio Social (SIASS) en un ambiente web. Esta herramienta tecnológica permitiría la simplificación de los procesos administrativos y operativos para el registro, renovación, evaluación y consulta de programas de servicio social, así como la emisión de la Carta Única de Liberación del Servicio Social de los estudiantes.

En este primer sistema web se involucró a todas las instancias participantes en el Proceso General de Servicio Social descrito anteriormente en este mismo capítulo, a fin de concentrar la información en una sola base de datos con la implementación de diferentes roles para consulta y registro de información de acuerdo a las características de cada instancia. Además se implementó un catálogo digital de consulta para todos los programas registrados en el que los alumnos tendrían acceso a un mayor número de opciones de servicio social de diferentes áreas multidisciplinarias.

PROBLEMA A RESOLVER

Actualmente el servicio social universitario está en una etapa de transición y redefinición de sus objetivos y de sus procesos, debido principalmente al cambio tecnológico que vivimos y al constante aumento de la población universitaria que deriva en volúmenes de información cada vez mayores.

El Sistema de Información Automatizada de Servicio Social que hoy en día se utiliza a nivel institucional tiene áreas de oportunidad para mejorar el desempeño en tiempo de respuesta y facilidad de uso para los usuarios administrativos, para los prestadores de servicio social y para las instituciones receptoras.

Algunos de los problemas que este sistema presenta son tecnología (hardware y software) obsoleta dado el tiempo que ha transcurrido desde su implementación, escasez de transparencia en los procesos y dificultad de acceso a la información pública relacionada con el Servicio Social Universitario que es de interés para los involucrados.

Lo anterior deriva en que los prestadores que buscan realizar el servicio social no tengan conocimiento de la gran oferta de programas de servicio social que existen en toda la Universidad y que no conozcan los lineamientos generales para iniciar esta actividad; además la naturaleza de los procesos actuales genera una considerable cantidad de trámites, los cuales retrasan el inicio de las actividades de los prestadores.

Adicionalmente la falta de estandarización en el desarrollo y documentación de la primera versión del SIASS ha ocasionado diversos problemas, muy conocidos en el ámbito del software, como son: mayor complejidad al hacer mantenimiento preventivo y correctivo, que usuarios y/o administradores que no estuvieron involucrados en el desarrollo no conozcan el funcionamiento del sistema y/o los procesos detalladamente, que no exista una metodología de respuesta a errores del sistema y desde luego la imposibilidad de crear actualizaciones sobre el sistema con la adición de nuevas reglas de negocio o módulos funcionales.

En términos administrativos, el manejo del sistema actual genera sobrecarga de trabajo debido a la centralización de los procesos y a la falta de automatización de los mismos, también dificulta la administración de la información y la entrega oportuna de reportes estadísticos y de resultados debido a la gran cantidad de datos que no pueden ser procesados automáticamente.

Con esto surge entonces la necesidad de redefinir también los sistemas de información del Servicio Social Universitario con la finalidad de optimizar los procesos, incluir las nuevas necesidades y permitir que lo anterior sea ejecutado por

todos los involucrados de una forma clara y sencilla; además de proporcionar información actualizada de manera oportuna.

TIPO DE ESTUDIO (PROPUESTA METODOLÓGICA)

Esta investigación se plantea basada en la metodología deductiva que utiliza hechos generales comprobados para establecer una hipótesis sobre situaciones particulares.

Como menciona Mario Tamayo en su obra *El Proyecto de Investigación* (1999, pág. 84), "... El procedimiento que consiste en desarrollar una teoría empezando por formular sus puntos de partida o hipótesis básicas y deduciendo luego sus consecuencias con la ayuda de las subyacentes teorías formales se llama método hipotético-deductivo".

De esta forma, el proyecto de investigación incluye los Elementos del Proceso Investigativo que Tamayo propone en el capítulo 1 de esta misma obra como son los antecedentes del problema o caso de estudio, el problema de la investigación, los objetivos, un marco teórico y las hipótesis que plantearemos para nuestro proyecto.

En nuestro caso plantearemos una hipótesis basada en el éxito que en la actualidad tienen la mayoría de los sistemas de información que son implementados con la finalidad de optimizar procesos y reducir costos de operación en las instituciones.

OBJETIVO GENERAL

Implementar un nuevo sistema de información basado en las nuevas estrategias y lineamientos del Servicio Social Universitario que cumpla con el propósito de optimización de los procesos y que a su vez mejore y acelere la funcionalidad del sistema de información actual.

Objetivos particulares

1. Proporcionar, a los responsables de escuelas y facultades, a las instituciones receptoras y al personal administrativo del Servicio Social Universitario, una herramienta que les facilite y optimice la realización de cada uno de sus respectivos procesos.
2. Implementar una interfaz que permita, a los alumnos de la Universidad, la consulta eficaz y oportuna de los diferentes programas de servicio social que se encuentran registrados.
3. Crear una herramienta que genere reportes y que sea capaz de obtener y procesar los datos del sistema de información de servicio social para traducirlos en información estadística útil para los responsables de escuelas y facultades y personal administrativo del servicio social universitario.
4. Establecer lineamientos de comunicación, bidireccional y sistematizada, con otras dependencias encargadas de dar seguimiento al avance académico de los estudiantes en la Universidad.

JUSTIFICACIÓN

El desarrollo de este proyecto surge a raíz de que se detectó la creciente necesidad de renovar y optimizar los procesos actuales del Servicio Social Universitario (SSU), estas necesidades se originan a partir de varios aspectos que han ido cambiando desde aquél 2007 donde se implementara por primera vez el Sistema de Información Automatizada de Servicio Social.

Si bien, con la implementación de ese sistema se lograron avances muy importantes y significativos en términos de simplificación de procesos, el paso del tiempo, el constante aumento en la población estudiantil y la evolución de las tecnologías de la información han generado estas nuevas necesidades que no han sido resueltas y que deben atenderse para dar respuesta a las exigencias actuales del Servicio Social Universitario y permitir que los procesos sigan mejorando, no solo para la DGOSE sino también para las nuevas necesidades y exigencias de la Universidad en general.

El sistema involucrará nuevamente a todos los usuarios que interactúan en el proceso del SSU, es decir, a los usuarios administrativos de la DGOSE, a las Unidades Responsables de Servicio Social (URSS) de escuelas y facultades, a las Instituciones Receptoras (IR) y a los estudiantes que realizarán servicio social. Cada uno con funciones y tareas específicas relacionadas con el rol que desempeñan dentro de los diferentes procesos del SSU.

Una vez que el sistema comience a funcionar, algunos beneficios comenzarán a vislumbrarse principalmente en el registro de instituciones receptoras y de programas de servicio social, estos procesos serán automatizados y se ejecutarán de una manera más rápida, agilizando los tiempos de captura, de lectura y de aprobación de los programas de servicio social.

Los beneficios en cuestión de optimización se extienden a cada una de las etapas del proceso general y se maximizan con la implementación de nuevas funcionalidades que no tiene el SIASS actual, por ejemplo el uso de la herramienta tecnológica de Firma Electrónica Avanzada que permitirá agilizar los tiempos de notificación de la liberación del servicio social para todas las instancias interesadas; en una era donde la ecología y la naturaleza se deben preservar, se ahorrará todo el papel y tinta utilizados para la impresión de cartas de liberación.

Adicionalmente este nuevo sistema permitirá generar gráficas e informes estadísticos personalizados de acuerdo a los intereses de cada una de las URSS y de la DGOSE. A pesar de que este nuevo sistema en su totalidad aumenta el valor de la DGOSE como dependencia reguladora del servicio social, la funcionalidad de estadística aumenta el valor que los usuarios le dan al propio sistema ya que les facilita y agiliza el acceso a la información concreta y procesada que ellos mismos generan pero que actualmente no pueden interpretar para su propio beneficio.

Resumiendo lo anterior, los beneficios que traerá la implementación de un nuevo sistema son los siguientes:

- Dar lugar a la implementación de las nuevas necesidades y reglas de negocio que tiene el SSU.
- Optimizar los tiempos de respuesta y disminuir la sobrecarga de trabajo para los trámites de registro de Instituciones Receptoras, programas de servicio social y alumnos, así como de notificación y certificación de término y liberación del servicio social.
- Facilitar la manipulación de los datos contenidos en el sistema para generar reportes estadísticos.
- Permitir el acceso a consultas de información específica para cada uno de los involucrados.

De acuerdo con lo descrito se considera que de seguir por la misma línea de trabajo, el sistema actual podría alcanzar su máxima capacidad de almacenamiento y ejecución en el corto plazo, lo cual derivaría en la pérdida de información y, en el peor de los casos en la inhabilitación definitiva del sistema; todo esto originado por la falta de mantenimiento debido a la nula documentación y al desconocimiento de la construcción del propio sistema. Adicionalmente los procesos del SSU seguirán siendo lentos y no podrían implementarse las nuevas necesidades que requiere la Universidad para satisfacer las demandas ante el constante desarrollo tecnológico y su creciente población estudiantil.

HIPÓTESIS

- H1: Si se implementa un sistema de información que utilice los procesos y lineamientos vigentes, utilizando tecnología actualizada se tendrá una mejor administración de la información de los prestadores, instituciones y programas registrados de servicio social en la Universidad.
- H2: Si se integran los datos de los programas registrados del servicio social vigentes en la Universidad, se brindará facilidad de acceso y transparencia a la información para los prestadores.

MARCO DE REFERENCIA

De acuerdo con el Plan de Estudios del año 2006 de la Carrera de Ingeniería en Computación (Facultad de Ingeniería. Universidad Nacional Autónoma de México (UNAM)., 2006) se determina que las materias más relevantes para el desarrollo de esta tesis son las siguientes:

- Álgebra
- Álgebra lineal
- Computación para ingenieros
- Programación avanzada y métodos numéricos
- Algoritmos y estructura de datos
- Probabilidad y estadística
- Sistemas operativos
- Ingeniería de software
- Lenguajes formales y autómatas
- Lenguajes de programación
- Ética profesional
- Bases de datos
- Administración de proyectos de software
- Redes de datos
- Administración de redes
- Costos y evaluación de proyectos
- Temas selectos de ingeniería de software
- Negocios electrónicos
- Verificación y validación de software

A continuación se menciona de manera breve la razón, en relación a los diferentes temas, de cada una de las asignaturas empleadas para elaborar este proyecto. Ya que sin la base de conocimientos adquirida en la carrera, hubiera sido muy difícil realizar este proyecto.

Álgebra

Parte del objetivo de la asignatura de Álgebra es la resolución de sistemas de ecuaciones para que permitan el estudio de la matemática aplicada (Consejo Académico del área de las Ciencias Físico Matemáticas y de las Ingenierías).

La resolución de ecuaciones nos ayudan a formar un pensamiento enfocado a la solución de problemas aplicados, es decir problemas con aplicaciones matemáticas en otros campos (Swokowski & Cole, 2006).

Por ejemplo en ésta disciplina se usan símbolos o letras para denotar números (Swokowski & Cole, 2006) que ayudan a la comprensión de un problema. Otro tema de valor es el de la teoría de conjuntos, que ayuda para entender la lógica de las bases de datos y sus relaciones, en pocas palabras saber álgebra es indispensable para poder realizar proyectos de programación.

Álgebra lineal

En general ayuda a lograr una abstracción tanto de ideas como de formas de resolver problemas. En computación se requiere, en gran medida, abstraer un problema, es decir separar las propiedades de un objeto a través de una operación mental, dejar de prestar atención al mundo sensible para centrarse en un pensamiento (Definición de Abstracción, s.f.). Para programación el ingeniero necesita abstraer un problema del mundo real y traducirlo de cierta forma que pueda crear una solución por medio de un sistema.

Computación para ingenieros

Esta materia resultó de suma importancia ya que nos adentró a los conocimientos básicos y fundamentales acerca de la importancia de la computación en el campo profesional. Las bases adquiridas en ésta materia permiten la elaboración de algoritmos, que es la primera aproximación que uno como ingeniero debe tener al elaborar un sistema.

Programación avanzada y métodos numéricos

En ésta materia uno de los temas que se estudia es la programación orientada a objetos, que es una forma de programación en la que se puede resolver un problema interpretándolo en conceptos como objetos que podamos manipular.

Algoritmos y estructura de datos

Principalmente en esta materia se estudian datos, su almacenamiento en estructuras, la recuperación y el ordenamiento de éstos. Puesto que en el desarrollo de éste sistema se usan datos, el uso y explotación de ellos se puede hacer con los conocimientos adquiridos en ésta materia.

Probabilidad y estadística

La probabilidad y estadística ayudan en el análisis de experimentos, al tener resultados de las ocurrencias de los eventos sucedidos en el experimento. Para ver si ésta tesis trae beneficio se deben analizar los resultados obtenidos. La estadística aporta conceptos que se utilizan como datos y observación que son el resultado de realizar un experimento.

Sistemas operativos

Conocer un sistema operativo permite saber cómo utilizarlo y a un nivel de administración, permite sacarle el mayor provecho porque se puede administrar la memoria y los procesos. En ésta materia también se vio programación y seguridad, eso nos entrenó para seguir fortaleciendo las habilidades de programación.

Ingeniería de Software

El objetivo de la materia señala que “El alumno aplicará en un proyecto procesos y herramientas mediante las cuales se analiza, diseña e implementa un proyecto de software” (Consejo Académico del Área de las Ciencias Físico Matemáticas y de las Ingenierías), y por lo tanto es la línea de referencia principal que se sigue para llevar a cabo el desarrollo de esta tesis.

En los temas de esta materia que se utilizaron está incluido el Lenguaje de Modelación Universal (*UML* por sus siglas en inglés) para diseñar la arquitectura del sistema, la planeación del sistema de programación, la toma de requerimientos del sistema de software y el diseño para estructurar el sistema web. Finalmente se usó el tema de validación y verificación del funcionamiento general del sistema de software.

Lenguajes formales y autómatas

Los conocimientos adquiridos en este curso pudimos homologarlos en la planeación del sistema al momento de elaborar los diagramas de clases y de procesos, para describir como debe ser el curso de nuestro sistema y los distintos caminos que puede seguir de acuerdo a las entradas o salidas de datos que se presentan en cada proceso.

Lenguajes de programación

Se aplicaron los conocimientos que dejó al cursar esta materia, en relación a los lenguajes de programación y paradigmas que involucran. Para este proyecto resultó fundamental el estudio de PHP como su sintaxis, semántica y cómo programar en éste lenguaje.

Ética profesional

El objetivo de la materia es muy claro: “El alumno fortalecerá su vocación humana y profesional en la conciencia de la dignidad, el deber y el ejercicio de la libertad, a través del conocimiento de la ética y el entendimiento de las responsabilidades”. (Consejo Académico del área de las Ciencias Físico Matemáticas y de las Ingenierías).

A lo largo del desarrollo de un trabajo cualquiera que sea éste, debe implicar una lista de valores morales que ayuden a un ejercicio sano de la profesión, como son la honestidad, el optimismo, la tenacidad, lealtad, entre otros. Silva Camarena explica que el trabajo implica un profundo compromiso con nuestro propio ser, y por lo tanto con los demás, que hacer bien lo que se hace es un compromiso moral con nosotros mismos, con nuestro quehacer, con nuestro gremio y nuestros semejantes (Camarena, 2000).

Bases de datos

La materia de base de datos es fundamental para el desarrollo del sistema web planteado en ésta tesis, ya que proporciona todas las herramientas y conocimientos generales que permiten diseñar, estructurar e implementar una base de datos Entidad-Relación para el almacenamiento de los datos requeridos por el mismo sistema.

Administración de proyectos de software

Los conocimientos adquiridos durante este curso se implementaron con el fin de llevar a cabo una buena planeación para el proyecto y así optimizar los recursos humanos, monetarios y de tiempo. Se utilizó el temario como base para definir un modelo que permita asegurar que el proyecto se finalice en tiempo y forma con la cantidad de recursos estimados y asegurando la calidad del mismo a través de la realización de pruebas.

Redes de datos

Debido a que se accederá al sistema desde internet, se necesita conocer términos como dominio, la tipología de red en que se implementará y otros, para que los usuarios puedan acceder al sistema.

Administración de redes

Es beneficioso conocer la administración de una red, lo que implica su planeación, organización, integración, dirección y control; para poder contar con un sistema de seguridad en la red y de acceso al sistema.

Costos y evaluación de proyectos

A pesar de ser una materia optativa el haberla cursado ayudó para poder dimensionar económicamente el tamaño de este proyecto, saber cuánto debe costar el desarrollo y la implementación y por qué. Aprender a costear y evaluar los proyectos en el aspecto monetario sin duda es algo que todos los estudiantes de la facultad deberíamos saber.

Temas selectos de ingeniería de software

Otra materia optativa que tomó relevancia ya que nos permitió conocer más a fondo temas específicos de la Ingeniería de Software como lo fue la construcción de Arquitecturas de Software que permiten describir un sistema de manera gráfica y profunda con información clara y puntual.

Negocios electrónicos

Mediante la inteligencia de negocios se hace notar cómo el desarrollo de software en los medios electrónicos aumenta el valor de los procesos administrativos y productivos de cualquier organización, optimizando los recursos y dejando una mayor satisfacción en los clientes que utilizan este tipo de desarrollos.

Verificación y validación de software

Como parte esencial de un buen desarrollo de software se debe incluir la verificación y validación dentro de la planeación del proyecto, para lo cual es necesario seguir metodologías de verificación y validación para la ingeniería del sistema, el desarrollo del software y para la aplicación de pruebas.

Ingeniería de Software

Las industrias, particulares o dependencias del gobierno hoy en día hacen uso del software para agilizar sus operaciones, administración, tener mejor control de la información o necesidades concretas que tengan y necesiten soluciones tecnológicas.

Según Sommerville (2005, p. 5) la definición de software son “programas de ordenador y la documentación asociada, los productos de software se pueden desarrollar para algún cliente en particular o para un mercado general”.

Fritz Bauer, citado por Pressman (2010, p. 11) tiene una definición de ingeniería de software que sirve de base para entender el concepto, “la ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales”.

Existe otra definición que el Instituto de ingenieros en electricidad y electrónica (IEEE) citado por Pressman establece que “la ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software” (Pressman, 2010, p. 11).

Modelos del Proceso de Software

Existen tres modelos generales o paradigmas del desarrollo del software: enfoque en cascada, desarrollo iterativo y basado en componentes (Sommerville, 2005, p. 61). Existen más modelos o formas de construir el software, pero se explican los procesos generales a continuación:

Cascada

Sólo se debe usar cuando los requerimientos se comprendan bien y sea improbable que cambien durante el desarrollo del sistema. A este modelo también se le conoce como ciclo de vida del software. El proceso no es rígido, ni un modelo lineal, sino que implica iteraciones en las actividades de desarrollo, ya que conforme se avanza una a una, se descubren deficiencias en alguna etapa anterior. (Sommerville, 2005, p. 62-63).

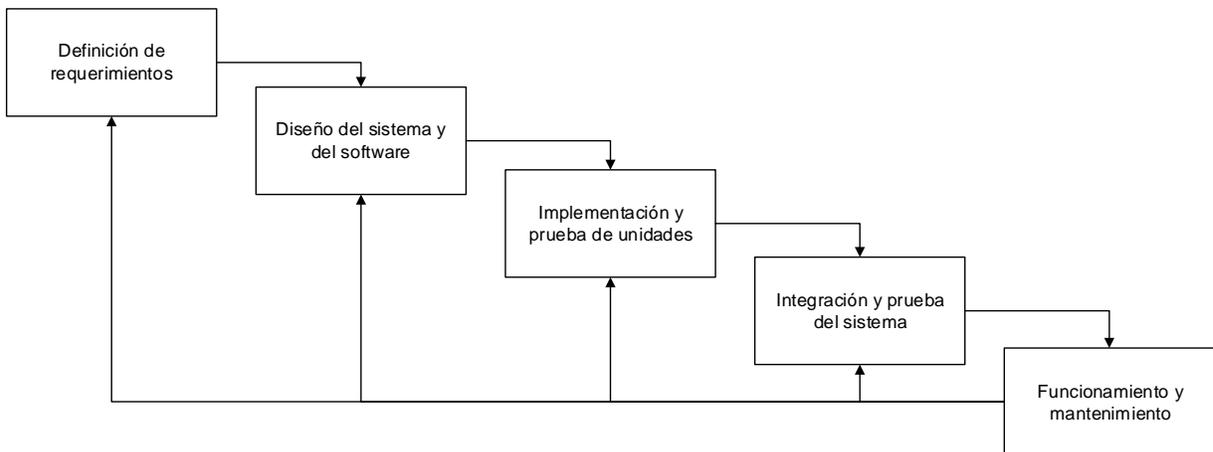


Ilustración 2 El ciclo de vida del Software. Fuente: adaptado de Sommerville (2005)

Las actividades fundamentales de este modelo son:

- Análisis y definición de requerimientos.

Se sostienen pláticas con el usuario para especificar los servicios, restricciones y metas del sistema.

- Diseño del sistema de software

En el diseño se implica la parte de hardware y la parte lógica (el software), porque se establece una arquitectura completa del sistema. Se pueden elegir que patrones de diseño se utilizarán, así como la parte física de en dónde se colocará la aplicación para que funcione.

- Implementación y prueba de unidades

En el desarrollo se crean conjuntos o unidades de programas, que tienen que ser verificadas, es decir que cumplan con la especificación inicial. Por lo general en el desarrollo hay un equipo de desarrolladores que crean el código y cada uno prueba la parte o unidad que se le asignó.

- Integración y prueba del sistema

Cuando todas las unidades han sido completadas y verificadas, se procede a integrarlas para que todas las partes construyan el todo y se pruebe como un sistema completo. Después de las pruebas se entregan al cliente.

- Funcionamiento y mantenimiento

Es cuando el sistema se ha entregado al cliente, instalado y está en funcionamiento. En el uso pueden ocurrir errores no detectados en el desarrollo y pruebas que necesitan corrección, es ahí cuando se da mantenimiento a los sistemas, a fin de corregir tales errores. También se da mantenimiento cuando surgen nuevos requerimientos que se necesita que el sistema realice, o bien se puede dar mantenimiento cuando se mejora la implementación del sistema.

Las ventajas de éste modelo son que la documentación se produce en cada fase y las desventajas presentadas son que en la documentación y aprobación se puede invertir mucho tiempo y por lo tanto los costos aumentan, ese tiempo hace que en ocasiones algunas actividades se queden en espera para terminarse y saltarse a otra actividad, por ejemplo, la especificación puede no completarse pero el desarrollo ya se está realizando, lo que puede provocar que al final el sistema esté mal estructurado y le falten requerimientos no considerados. Como se mencionó antes, para este modelo se necesita conocer bien la especificación al inicio, pero eso suele ser difícil porque el cliente no siempre tiene en claro explícitamente cada uno de los requerimientos, por lo que este modelo es poco flexible a cambios. Otro inconveniente es que como se podrá ver el resultado ya en una etapa muy avanzada, sí hay algo que se entendió mal, o un error grande no se detectó, eso podría afectar el resultado final.

Evolutivo

En ocasiones, por el tipo de requerimiento, o negocio, o porque el cliente no tenga claramente definido lo que necesita o algún factor; se requiere un producto pronto para satisfacer las necesidades inmediatas, que sea funcional y que se pueda poner en marcha lo más pronto posible, y no se puede esperar a la larga línea del proceso del ciclo de vida normal. Por ello surge el modelo evolutivo que permite que se generen con rapidez productos de trabajo incremental (versiones funcionales del software). Cada vez se entregan versiones mejoradas y más completas del sistema. Éste modelo está diseñado para aceptar los cambios, porque hay versiones continuas. (Pressman, 2010, p.36-38).



Ilustración 3 Desarrollo evolutivo. Fuente: adaptado de Sommerville (2005).

Para un desarrollo evolutivo se pueden hacer prototipos, experimentando con los requerimientos del cliente, para ayudarlo a comprender lo que requiere cuando los requerimientos no están claros. Para hacer un prototipo se necesitan los objetivos

generales del software, aunque no se cuente con los detallados, se hace un modelado y se construye un prototipo que se presenta al cliente, quien da una retroalimentación y va definiendo de mejor manera los requerimientos, eso ayuda también a comprender mejor las reglas de negocio. Una desventaja es que como se entrega un prototipo se tiene que volver a rehacer algunas o muchas partes

El desarrollo exploratorio sirve para un modelo evolutivo, en éste se trabaja directamente con el cliente para estudiar sus requerimientos, se comienza a desarrollar las partes que mejor se entienden y así ir agregando los requerimientos que el cliente proporciona. (Sommerville, 2005)

Las ventajas de un desarrollo evolutivo es que pronto se desechan incertidumbres en la especificación y que se satisface pronto las necesidades próximas del cliente, la desventaja es que el producto que se entrega es limitado, que se rehace muchas veces, y la calidad puede estar comprometida por la pronta entrega de una versión.

Componentes

Es un desarrollo que se basa en la reutilización de componentes de software reutilizable, a veces estos componentes son sistemas por si mismos que pueden ofrecer una funcionalidad específica, así se construye un sistema a partir de fragmentos de software prefabricados. (Pressman, 2010, p. 43).

Las actividades que se llevan a cabo son: el análisis de componentes, según el requerimiento se busca el componente más adecuado que supla cierta funcionalidad, considerando los aspectos de integración de los componentes, si son factibles en el sistema, ya sea por costo, compatibilidad, versiones u otros; diseño de una arquitectura para recibir los componentes; integración de componentes y ejecución de pruebas para asegurar la funcionalidad completa. (Sommerville, 2005, p. 65).

Como ventaja de este modelo es que se reduce el tiempo de desarrollo. Y como desventaja puede ser que el sistema no cumpla las necesidades, porque algún componente no era el adecuado.

Proceso Unificado de *Rational* (RUP)

Pressman (2010) menciona que “el proceso unificado es un intento por obtener los mejores rasgos y características de los modelos tradicionales del proceso de software, pero en forma que implemente muchos de los mejores principios del desarrollo ágil del software”.

El Proceso Unificado proporciona un enfoque de disciplina, asignación de tareas, y responsabilidades en una organización, con ello se mejora la productividad del equipo, porque se siguen directrices, plantillas en una base de conocimiento, de esa manera no importa el área en que cada miembro del equipo se desempeñe, como

requisitos, diseño u otros, todos comparten un lenguaje común, un proceso y una vista de cómo desarrollar el software. (Rational software, 1998)

RUP interactúa con el Lenguaje de Modelado Unificado (*UML*) para poder representar la vista de cómo se desarrollará el sistema, eso permite a los integrantes conocer cómo se hará el sistema, independientemente del o los lenguajes de programación que se utilicen.

Las fases de *RUP* son:

1. Inicio

Se interactúa con el cliente para conocer los requerimientos del negocio. Se propone una arquitectura tentativa. Y se hace una planeación para identificar los recursos que se necesitan, evaluar riesgos y se definen las medidas que se van a tomar mientras avanza el incremento del software. (Pressman, 2010, pp. 46-47)

2. Elaboración

Se analiza el problema, se establece la arquitectura y el plan de desarrollo. Es la fase más crítica de las cuatro fases. Al terminar esta fase se tiene un modelo de requerimientos en casos de uso, una descripción de la arquitectura y un prototipo de ésta, una lista de riesgos evaluados y un plan de desarrollo. (Rational software, 1998, pp. 4-5)

3. Construcción

Los componentes y características de la aplicación son programados e integrados en el producto, utilizando la arquitectura que se hizo en la elaboración. A medida que se desarrolla, se hacen pruebas unitarias y se hacen tareas de integración (Sommerville, 2011, p. 77). Al terminar esta fase se debe tener el producto de software terminado para las plataformas requeridas, manuales de usuario y una descripción de la versión que se entrega (Rational software, 1998, p. 6).

4. Transición

Es mover el producto de software a un entorno real con los usuarios, quienes prueban el sistema y pueden reportar defectos y cambios necesarios, para el desarrollo de nuevas versiones. También se capacita a los usuarios. Al finalizar se tiene un software funcionando correctamente (Rational software, 1998, pp. 6-7).

Las fases de *RUP* no ocurren secuencialmente, sino en forma escalonada e iterativa, es decir, “cada fase se puede representar de un modo iterativo con los resultados desarrollados incrementalmente” (Sommerville, 2005, p. 77).

La ventaja de *RUP* es que es genérico, utiliza *UML* que es entendible para todos los miembros del equipo, permite iteraciones en las fases.

Los modelos anteriores son modelos tradicionales, cada uno tiene un flujo distinto para el desarrollo de software. Sin embargo la dificultad común que se presenta en

ellos, es que son poco flexibles a los cambios, ya que, aun cuando sean procesos iterativos, se debe esperar al inicio de la próxima iteración para incluir los cambios, y ese período de tiempo puede ser variable, tanto puede ser poco como puede extenderse por un largo período. A pesar de ello si puede haber proyectos en los que se tiene claramente definidos los requerimientos al inicio, y en los cuales se puede utilizar las metodologías tradicionales.

Proceso del Software

Para cumplir con una meta determinada es necesario tener en cuenta las actividades que se tienen que realizar para que pueda realizarse, de la misma manera, la construcción de software requiere de un proceso definido para llevarlo a su fin, se necesita conocer los procesos que se involucran en el desarrollo, mantenimiento y operación del software.

“Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo” (Pressman, 2010, p. 12), “un proceso define quién hace qué, cuándo y cómo para alcanzar cierto objetivo” (Jacobson I. , 1992). Como cada proyecto se utiliza en diferentes escenarios (por mencionar algunos hay de comercio electrónico, de tiempo real, etc.), el proceso no puede ser el mismo para todos, por ello las personas involucradas pueden elegir el conjunto apropiado de acciones y tareas para el trabajo, teniendo en cuenta que al elegir el proceso adecuado para ese software, éste se entregue de forma oportuna y con calidad.

Existen cuatro actividades fundamentales comunes a los proceso del software (Sommerville, 2005):

- Especificación del software: define lo que se va a producir y sus restricciones, se busca entender los objetivos del proyecto, y reunir los requerimientos para definir las características y funciones del software.
- Desarrollo del software: el software se diseña y se programa, se crean modelos o “bosquejos” para poder entender de una manera general lo que se requiere y que diseño lo satisface, posteriormente se procede a generar el código.
- Validación del software: el software se válida, es decir se revisa si cumple lo que el cliente requiere y solicitó.
- Evolución del software: el software se modifica para adaptarlo a los cambios que el cliente solicita o a las necesidades que surjan en las reglas de negocio.

Existen estándares desarrollados por reconocidas organizaciones, que establecen métodos y formas de trabajo para hacer una buena ingeniería de software. El

Instituto de Ingenieros en Electricidad y Electrónica (IEEE, por sus siglas en inglés) ha desarrollado estándares para las áreas de Ingeniería de Software. Algunos de ellos describen las actividades mencionadas anteriormente, como los son el IEEE Estándar 830 Prácticas recomendadas para las especificaciones de software, IEEE Estándar 1016 Descripción del diseño del software, IEEE Estándar 1012 para la verificación y validación del software y el IEEE Estándar 1219 para el mantenimiento del software.

Se revisarán estas actividades fundamentales para una mejor comprensión de los modelos del proceso de software descritos posteriormente.

Requerimientos del Software

Al momento que inicia un proyecto es necesario conocer lo que se quiere que el sistema haga, los clientes deben definir sus necesidades y sus restricciones.

La Guía al Cuerpo de Conocimiento de la Ingeniería de Software (por su nombre en inglés *Guide to the Software Engineering Body of Knowledge, SWEBOK Guide*) nos proporciona una definición de lo que es un requerimiento de software: “un requerimiento de software es una propiedad que debe ser exhibida como algo en orden para resolver algún problema en el mundo real” (IEEE Computer Society, 2014, pp. 32-33). El estándar 610.12-1990 de la IEEE da como definición de requerimiento que es “una condición o habilidad necesitada por un usuario para resolver un problema o lograr un objetivo. Una condición o capacidad que debe cumplir o poseer un sistema o un componente de sistema para satisfacer un contrato, estándar, especificación u otros documentos impuestos formalmente” (IEEE Computer Society, 2005, p. 114).

El SWEBOK define algunos pasos claves para el proceso de ingeniería de requerimientos (IEEE Computer Society, 2005, pp. 113-119):

- **Obtención de requerimientos**

La información necesitada para desarrollar y documentar es obtenida de las partes interesadas para descubrir sus necesidades, identificar y negociar posibles conflictos, y establecer un marco claro y límites para el proyecto. Al iniciar esta fase se debe comprender el problema que se desea resolver y el ambiente en el que se propone que la aplicación sea ejecutada. Los interesados deben hacer un caso de negocio, es decir, un análisis más completo entendiendo los costos, riesgos y anticipando beneficios del proyecto.

Existen técnicas que ayudan a obtener los requisitos, desglosando el problema en unidades pequeñas y más entendibles. Una de esas técnicas son las sesiones colaborativas entre los participantes, donde la tarea del ingeniero de requerimientos es identificar las áreas o puntos en que los participantes estén de acuerdo, y las tareas que estén en conflicto en la que no todos estén de acuerdo

y mediar para llegar a algo en común. También se pueden tener entrevistas, cuestionarios, estar en el lugar del cliente o del negocio para observar el entorno, y prototipos.

- **Análisis de requerimientos**

Implica obtener una comprensión más profunda para el producto y sus interacciones, identificando el impacto de los requerimientos en un nivel de diseño, e identificando algunos conflictos que pueden surgir. En éste análisis se puede clarificar la clasificación de los requerimientos que se tienen.

Los requerimientos pueden clasificarse en funcionales y no funcionales. Los requerimientos funcionales describen lo que el sistema debe hacer y en ocasiones lo que no debe hacer, describen con detalle la función de éste, sus entradas, salidas y excepciones. Los requerimientos no funcionales son “restricciones de las funciones ofrecidas por el sistema, restricciones de tiempo, sobre el proceso de desarrollo y estándares” (Sommerville, 2005, p. 110).

En el análisis se considera la prioridad de los requerimientos, donde los de alta prioridad son más esenciales como metas; el alcance de los requerimientos, que se refiere a la manera en que afecta los requerimientos al software o a los componentes del mismo; la estabilidad de los requerimientos ya que durante el ciclo de vida pueden cambiar. (IEEE Computer Society, 2014, p. 39)

- **Documentación de requerimientos**

La especificación del sistema, consiste de una serie de documentos que describen los requerimientos de una manera adecuada para usarlos en el diseño, mantenimiento, evaluación y aprobación del software. El mantenimiento de los requerimientos comienza una vez que se inician a recabar, ya que pueden surgir cambios, por lo que es necesario llevar un control de cambios del documento de especificación de requerimientos.

La especificación de requerimientos es un documento que describe el sistema para ser desarrollado en un formato que puede ser revisado, evaluado y aprobado. Éste documento va dirigido a varias personas, al dueño o cliente del sistema, administradores, y hasta desarrolladores, por lo tanto el nivel de detalle debe ser cuidadoso y libre de ambigüedades, para que los desarrolladores comprendan claramente lo que se necesita hacer, evitando realizar los menos errores posibles, ya sea en desarrollo de nuevas funciones o en mantenimiento del sistema.

- **Validación de requerimientos**

Utiliza el modelado, revisiones, prototipos y las pruebas de aceptación del producto final para verificar y validar que los requerimientos estén correctos.

El Estándar 830-1998 de la IEEE Prácticas Recomendadas para la especificación de requerimientos de software (*Recommended Practice for Software Requirements Specifications*, comúnmente llamado “SRS”) ayuda a definir el documento que se elabora con todos los requerimientos del proyecto. La estructura del escrito que presenta el estándar es la siguiente (IEEE Computer Society, 2005, pp. 176-180):

1. Introducción
 - 1.1 Propósito
 - 1.2 Alcance (describe la aplicación en específico, incluye los beneficios, objetivos y metas)
 - 1.3 Definiciones, acrónimos y abreviaturas
 - 1.4 Referencias
 - 1.5 Descripción del resto del documento
 - 2 Descripción General
 - 2.1 Perspectiva del producto (debe poner el producto en perspectiva con otros productos relacionados, si el producto es independiente, o si es componente de otro sistema)
 - 2.2 Funciones del producto (debe proporcionar un resumen de las principales funciones que el software realizará)
 - 2.3 Características del usuario
 - 2.4 Restricciones (Como políticas regulatorias, limitaciones de hardware, seguridad, etc.)
 - 2.5 Suposiciones y dependencias (una lista de los factores que pueden afectar los requerimientos)
 - 3 Requerimientos Especificos (Debe contener los requerimientos a un detalle suficiente que permita diseñar un sistema que satisfaga lo solicitado, deben tener una descripción, entradas, salidas y funciones)
- Apéndices
Índice

Sin embargo Sommerville (2005, p. 125) hace notar que es un marco muy general, por lo que puede ser difícil que una organización lo adopte por completo, pero que si se puede adaptar según las necesidades.

Otra observación hecha por el autor es que en las metodologías ágiles los requerimientos cambian rápidamente, en cuyo caso el documento de requerimientos quedaría desfasado. Puesto que la gestión de requerimientos es el proceso de controlar los cambios en los requerimientos que los clientes necesiten en el sistema, entonces se documenta un control de cambios.

Diseño del Software

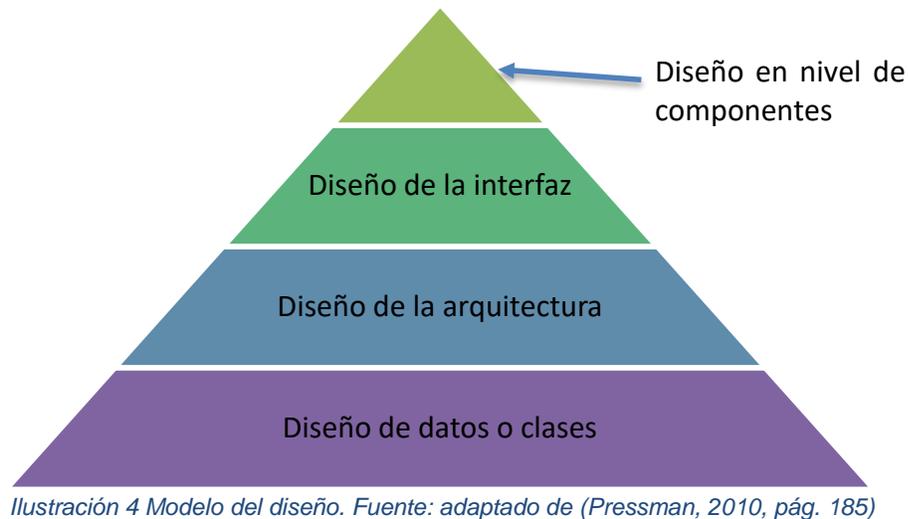
Posterior a la toma de requerimientos, se hace como una traducción de ellos para planear la forma de construcción en un nivel técnico. “El diseño crea una representación o modelo del software... proporciona detalles sobre la arquitectura del

software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema.” (Pressman, 2010, p. 183).

El *SWEBOK* menciona que “el diseño se define como dos, el proceso de definición de arquitectura, componentes, interfaces y otras características de un sistema o componente y el resultado de [ese] proceso. Visto como un proceso, el diseño de software es [una] actividad del ciclo de vida en el que los requisitos de software se analizan con el fin de producir una descripción de la estructura interna del software que servirá de base para su construcción” (IEEE Computer Society, 2014, p. 50)

Existen métodos y estrategias para poder comprender mejor el problema, el *SWEBOK* (IEEE Computer Society, 2014, p. 59) nos habla de algunas como lo son: el diseño orientado a funciones o estructurado, el diseño orientado a objetos, diseño de datos con estructura centrada y diseño basado en componentes. En términos muy generales el diseño orientado a funciones identifica las funciones que debe tener el software ordenándolos jerárquicamente, se elaboran diagramas de flujo y descripciones de procesos asociados. El orientado a objetos es una abstracción, donde el problema que se pretende resolver se traslade a términos de cosas donde un sustantivo es un objeto, un verbo es un método y un adjetivo es un atributo. El diseño de datos con estructura centrada, primero se describe la estructura de datos sus entradas y salidas, después se desarrolla la estructura de control del programa en base a esos diagramas. En el diseño basado en componentes es el desarrollo e integración de componentes con el fin de usar la reutilización, donde un componente de software es una unidad independiente que se puede desplegar de forma independiente.

El modelo de diseño muestra un nivel de abstracción donde los elementos del modelo se transforman en equivalentes de diseño y se van mejorando. En el modelo de diseño se usan varios de los diagramas UML Lenguaje de Modelado Unificado que se utiliza como modelo de análisis, estos diagramas se elaboran como parte del diseño (Pressman, 2010, pp. 197-198). Pressman considera el siguiente modelo de diseño con cuatro elementos, conforme se desarrolla cada uno surge una visión más completa del diseño.



El diseño de datos involucra diagramas de clases, con representación de objetos y sus relaciones, sirviendo de base para el diseño de datos. Se crea un modelo de datos que se irá refinando de forma progresiva, éste modelo estará altamente ligado con la aplicación, para el diseño de las estructuras de datos, así como de los algoritmos que tendrá el software, por lo que la base de datos es clave en una buena aplicación (Pressman, 2010, pág. 199).

El diseño de la arquitectura involucra diagramas de clases, diagramas de colaboración, de flujo de datos, narrativas de procesamiento. Este diseño “define la relación entre los elementos principales de la estructura del software, los estilos y patrones de diseño” (Pressman, 2010, pág. 185). Para su realización se usan estilos arquitectónicos, estructuras arquitectónicas, patrones de diseño y por supuesto hay una toma de decisiones (IEEE Computer Society, 2014, pp. 53-54).

El diseño de la interfaz traza la forma en que se interactúa con el sistema, ya sea del usuario al sistema o de otro sistema. Para esto ayudan los diagramas de estado, secuencia, de flujo de datos, flujo de control, diagramas de casos de uso y diagrama de actividades. Existen tres elementos que se deben considerar, la interfaz del usuario, las interfaces externas con otros sistemas y las interfaces internas que es con los diferentes componentes del diseño. “En pocas palabras, una interfaz es un conjunto de operaciones que describen alguna parte del componente de una clase y dan acceso a aquéllas” (Pressman, 2010, p. 200).

A nivel de diseño de componentes, esta es una descripción de los elementos de la estructura del software que expone los detalles internos de cada componente, es decir cada uno de los módulos (componentes) que constituyen la arquitectura. Comparándolo a un plano de una casa, es como si fueran los planos de cada habitación. Se usa el diagrama de UML de componentes para modelarlo, y para detalles del componente se puede utilizar por ejemplo el diagrama de actividades que muestra la lógica del procesamiento.

Para elaborar el documento, y considerando el nivel de detalle deseado, entre las muchas plantillas que existen para elaborar el documento de diseño se puede considerar la proporcionada por la *IEEE Computer Society* (2005, pp. 284-287)

1. Introducción
2. Diseño de vistas
3. Descripción de descomposición
 - 3.1 Alcance
 - 3.2 Uso
 - 3.3 Representación
4. Descripción de dependencias
 - 4.1 Alcance
 - 4.2 Uso
 - 4.3 Representación
5. Descripción de Interfaces
 - 5.1 Alcance
 - 5.2 Uso
 - 5.3 Representación
6. Descripción del diseño detallado
 - 6.1 Alcance
 - 6.2 Uso
 - 6.3 Representación

El estándar que sirve de apoyo para la documentación del diseño es el 1016-2009 de la IEEE, Descripción del Diseño de Software, comúnmente llamado SDD, el cual dice que el contenido requerido para un SDD es (*IEEE Computer Society*, 2009, p. 7):

- Identificación del SDD

Donde la identificación del SDD incluye información como el alcance, referencias, contexto, etc.

- Identificación de los interesados del diseño

En la identificación de los *stakeholders* del diseño se refiere a los interesados en el diseño del software, ya sean individuos, organizaciones o grupos.

- Identificación de las áreas de interés

Un *design concern* es un área de interés con respecto al diseño del software, por ejemplo algunos de los *design concern* son los servicios que ofrecerá el sistema, que usuarios actuarán en el sistema, la estructura de clases, etc.

- Diseño de los puntos de vista seleccionados

Para el diseño de los *viewpoints* o puntos de vista, cada uno identifica las áreas de interés en los que hará énfasis con sus vistas, la especificación de los elementos y acuerdos disponibles para la construcción.

- Diseño de vistas

Una vista es la representación de uno o más diseños de elementos que comprenden un conjunto de áreas de interés de un punto de vista específico. Cada diseño de vista es administrado o pertenece al diseño de un punto de vista; donde cada punto de vista pone atención en un conjunto de áreas de interés.

- Diseño de adicionales (*overlay*)

Un diseño de un overlay es para representar la información adicional respecto a algún diseño de vista ya definido.

Se muestra una tabla que resume los puntos de vista con las áreas de interés correspondientes y las vistas que se pueden usar en cada punto de vista.

Diseño de Punto de vista	Diseño de áreas de interés	Ejemplos de diseño de lenguajes
Contexto	Servicios del sistema y usuarios.	Diagrama uml de casos de uso.
El propósito es identificar en un diseño los servicios ofrecidos y sus actores. Describe los servicios provistos por un caso de diseño con referencia a un contexto explícito.		
Composición	Ensamblado de sistemas, subsistemas, componentes, reuso de componentes.	Diagrama uml de paquetes, diagrama uml de componentes, diagrama de despliegue.
Describe el diseño de la estructura del sistema en sus partes; esto ayuda a los desarrolladores y los que dan mantenimiento a identificar los componentes del sistema, para localizar las funcionalidades, responsabilidades y otros roles de los componentes. Es de ayuda para analizar el impacto de cambios.		
Lógico	Estructura estática (clases, interfaces y sus relaciones). Reuso de clases.	Diagrama uml de clases y diagrama de objetos.
Es para elaborar y diseñar las implementadoras de las clases e interfaces con sus relaciones.		
Dependencia	Interconexión y parametrización.	Diagrama uml de paquete, diagrama de componentes.
Especifica las relaciones de las interconexiones y accesos de las entidades (subsistemas, componentes y módulos). Estas relaciones pueden ser de compartir información, orden de ejecución o parametrización de interfaces.		
Información con distribución de datos y su volumen.	Información de persistencia.	Diagrama entidad-relación, diagrama uml de clases.

Es aplicada cuando la persistencia de datos es esperada en el sistema. Incluye la estructura de datos, su contenido, estrategias de mantenimiento y accesos.		
Patrones	Reuso de los patrones y los frameworks disponibles.	Diagrama uml de composición.
Se consideran las ideas de diseño como patrones de colaboración que envuelven roles y conectores abstractos.		
Interfaz	Definición del servicio y acceso al servicio.	Diagrama uml de componentes y definición de lenguaje de la interfaz.
Provee información a los diseñadores, programadores y testers acerca de cómo es el uso correcto del sistema para acceder a los servicios.		
Estructura	Componentes internos y la organización de los componentes y clases.	Diagrama uml de clases y diagrama de estructura.
Es usado para conocer las partes internas y organización del sistema.		
Interacción	Comunicación de los objetos y mensajes.	Diagrama uml de secuencia y de comunicación.
Se definen las estrategias para la interacción entre las entidades, acerca de cómo, cuándo y que acciones ocurrirán.		
Estados dinámicos	Transformación de estado dinámico.	Diagrama uml de máquina de estados, estados de transición.
Este punto de vista es para los que están interesados en el comportamiento interno del sistema.		
Algoritmo	Procedimiento lógico	Tablas de decisión.
Una descripción de las operaciones como métodos y funciones, los detalles internos y lógicos.		
Recursos	Utilización de recursos	Diagrama uml de clases, de objetos.
Es para modelar las características y utilización de los recursos en un sistema. Por ejemplo el uso de software como librerías, o de dispositivos físicos como impresoras, memorias, etc.		

Tabla 1 Resumen de viewpoint. Adaptado de (IEEE Computer Society, 2005)

Administración de Proyectos

Pressman (2010, p. 554) observa que “construir un software de computadora es una labor compleja, particularmente si involucra a muchas personas que trabajan durante un tiempo relativamente largo, por eso es necesario administrar los proyectos de software”. Es una actividad compleja y amplia, en la cual si se tiene una mala administración puede conducir al fracaso del proyecto. Por el contrario si se cuenta con una buena administración los resultados son favorables.

La Guía de fundamentos para la dirección de proyectos (PMBOK Guide, 2000, p. 6) expresa que la “gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del proyecto... El equipo del proyecto gestiona el trabajo de los proyectos, y el trabajo implica generalmente: demandas competitivas de alcance, tiempo, riesgo y calidad; actores con diferentes necesidades y expectativas; y requisitos identificados”.

Roger Pressman (Pressman, 2010, p. 555) dice que una buena administración contempla cuatro bloques, la administración del personal, producto, proceso y proyecto. Haciendo referencia a lo anterior del *PMBOK*, la administración también implica temas de planificación, calendarización, control de riesgos.

Para la administración del personal, es importante tener en claro que si bien las empresas de desarrollo de software crean un producto intangible, las personas que participan en el desarrollo son las que interactúan y colaboran para lograr el objetivo, por ello es importante considerarlos como pieza clave en el éxito del proyecto. “Toda organización requiere mejorar continuamente su habilidad para atraer, desarrollar, motivar, organizar y conservar la fuerza de trabajo necesaria a fin de lograr sus objetivos empresariales” (Pressman, 2010, p. 555).

Los actores que están involucrados en los proyectos son gerentes ejecutivos, gerentes de proyecto, clientes, usuarios, líderes de proyecto y equipo de trabajo. Los gerentes ejecutivos son los que definen cuestiones empresariales; los gerentes de proyecto aquellos que planifican, motivan, organizan y controlan a los que hacen el software; los clientes, los que especifican los requerimientos; los usuarios son los que interactúan con el software fabricado; el líder de proyecto debe motivar al personal y ser hábil en la resolución de problemas; en el equipo de desarrollo se debe considerar la dificultad del problema, el tamaño del producto, comunicación y análisis. El objetivo es contar con un equipo cohesionado, según DeMarco y Lister (Pressman, 2010, p. 560) los miembros de un equipo cuajado son más productivos, tienen una meta en común. Lo que hay que evitar son equipos tóxicos, llamados así por Jackman (Pressman, 2010, p. 560) en los cuales existe frustración, fricción entre los miembros y poca coordinación.

Para la administración del producto, el gerente del proyecto debe examinar, establecer y acotar el ámbito del producto. La primera actividad en la administración

es determinar el ámbito del producto que se solicita, tener claro los objetivos y restricciones (Pressman, 2010, p. 563). Los objetivos reflejan las metas globales para el producto, así como identificar el ámbito que es identificar los datos, funciones y comportamientos principales del producto. Teniendo claro lo anterior, se pueden considerar la o las soluciones a tomar.

Conociendo lo que es un proceso de software comentado anteriormente, se puede establecer un plan de trabajo para el desarrollo de software, donde algunas veces puede ser complicado elegir un modelo de proceso de software, ya que puede llevarse a cabo parcialmente, por lo que se debe ser flexible en adaptarlo y quizá combinarlo con otros modelos. Sin embargo hay algunas actividades comunes, independientemente de cual modelo de proceso se elija, estas actividades son la comunicación, planificación, modelado, construcción y despliegue del software.

Lo que se pretende en una buena administración del proyecto es alcanzar su éxito, Capers Jones (Pressman, 2010, p. 556) considera que un proyecto exitoso es el que cumple sus objetivos de calendario, costo y calidad, no todos los proyectos cumplen estas metas, por el contrario la mayoría tiene demoras o grandes demoras y peor aún se dan por terminados sin haber completado las tareas. Para evitar que el proyecto falle, debe haber una gran coordinación entre los ingenieros de software y los administradores del proyecto, donde se “desarrolle un enfoque de sentido común para planificar, monitorear y controlar el proyecto” (Pressman, 2010, p. 556).

Riesgos y Control de Riesgos

El control de riesgos debe estar contemplado en la administración de proyectos de software. El *SWEBOK* (IEEE Computer Society, 2014, p. 138) expresa que el riesgo es caracterizado por la probabilidad de un evento que puede resultar con un impacto negativo en un proyecto.

El manejo de estos riesgos comprende su identificación y análisis en base a su probabilidad y prioridad, con el fin de tener estrategias para mitigarlos y con ello reducir el impacto negativo que podría causar problemas.

Los riesgos pueden afectar al proyecto en cuanto a tiempos de entrega o recursos, de igual manera se puede perjudicar el producto respecto a su calidad y rendimiento, otro es el riesgo del negocio, el cual afecta a la organización que desarrolla. Pueden existir muchos factores que amenacen el proyecto como lo puede ser la rotación de personal, cambio de gestión de la organización, falta de disponibilidad de hardware con las especificaciones necesarias, subestimación del tamaño, en fin, pueden ser varios factores, que se podrían prevenir, por lo que los planes de contingencia se deben mantener actualizados.

Para iniciar la gestión de riesgos se debe hacer una identificación de éstos, puede ser basado en experiencia, en ideas, aunque Sommerville (2005, p. 98) hace notar al menos seis tipos de riesgos:

- Riesgos de tecnología, ya sean del software o hardware que se utiliza para el desarrollo.
- Riesgos de personal, asociados al personal que colabora.
- Riesgos organizacionales, el lugar de la organización donde se desarrolla el software.
- Riesgos de herramientas, pueden surgir del software de apoyo utilizado en el desarrollo.
- Riesgos de requerimientos, de los cambios del cliente y la manera en cómo se llevará a cabo dicho cambio.
- Riesgo de estimación, de cuestiones administrativas alrededor del sistema y de los recursos financieros para la construcción.

En segundo lugar se procede a un análisis de los riesgos identificados, es un proceso complicado porque no puede existir una certidumbre específica acerca de la probabilidad que ocurra, por lo que se hace una valoración en intervalos, la probabilidad puede ser muy baja, baja, moderada, alta o muy alta, con lo que se puede valorar si un riesgo es tolerable o es catastrófico. (Sommerville, 2005, p. 98).

Una vez valorados los riesgos de probabilidad e impacto, se debe determinar cuáles son los más importantes y hacer una planificación de ellos, se pueden tomar estrategias de prevención para evitar que ocurran, estrategias de minimización para que el impacto no sea de dimensiones catastróficas, y por último tener planes de contingencia en caso de que se presente el peor escenario.

Planificación de Proyectos de Software

La planificación es un conjunto de actividades que se hacen al comienzo de la administración del proyecto de software. Antes de iniciar el proyecto, se debe estimar el trabajo que se va a realizar, los recursos necesarios y el tiempo que se requerirá. Se hacen estimaciones de recursos, costo y calendario. Posteriormente se hace una planificación calendarizada, con tareas y los responsables de hacerlas.

Pressman (Pressman, 2010, p. 595) propone un conjunto de tareas para la planificación de proyectos:

1. Establecer ámbito del proyecto
2. Determinar la factibilidad
3. Analizar los riesgos
4. Definir recursos requeridos: determinar recursos humanos, definir recursos de software y hardware.
5. Estimar costo y esfuerzo
6. Desarrollar un calendario del proyecto: establecer conjunto de tareas.

El ámbito del software describe las funciones y sus características que se entregan al usuario final, las restricciones, los datos de entrada y salida, y el contenido que se

mostrará a los usuarios. El ámbito puede presentarse en una descripción y en casos de uso. Una vez aterrizado lo anterior es sano preguntarse si es factible o posible de realizarse, ya sea en cuanto a tecnología, costos y tiempo, se debe contestar con honestidad para saber si es aceptable de cumplirse o no, de no ser así tener una negociación y llegar a un acuerdo.

La estimación de costo y esfuerzo es una labor compleja, ya que influyen diferentes variables que pueden modificar el costo final del software y del esfuerzo que se requirió para su desarrollo (Pressman, 2010, p. 599).

Metodologías Ágiles

La definición de la palabra ágil es “que actúa o se desarrolla con rapidez o prontitud” (Real Academia Española, 2014). En la actualidad la agilidad es una cualidad que demandan las personas, y la creación de software también se requiere que se comporte de esa manera, las metodologías ágiles son otra forma de desarrollar software.

Ivar Jacobson observa lo siguiente:

“Todos son ágiles. Un equipo ágil es diestro y capaz de responder de manera apropiada a los cambios. El cambio es de lo que trata el software en gran medida. Hay cambios en el software que se construye, en los miembros del equipo, debidos a las nuevas tecnologías, de todas clases y que tienen un efecto en el producto que se elabora o en el proyecto que lo crea. Deben introducirse apoyos para el cambio en todo lo que se haga en el software; en ocasiones se hace porque es el alma y corazón de éste. Un equipo ágil reconoce que el software es desarrollado por individuos que trabajan en equipo y que su capacidad, su habilidad para colaborar, es el fundamento para el éxito del proyecto” (Jacobson I. , 2002).

Desde su punto de vista, el cambio está presente, por lo que se requiere una respuesta efectiva a los cambios.

“Los modelos de desarrollo *Agile* atienden las deficiencias asociadas con los modelos tradicionales de gestión de [proyectos] para satisfacer las crecientes demandas ambientales y expectativas que las organizaciones encaran. Dado que los modelos tradicionales de gestión de [proyectos] en general hacen hincapié en una amplia planificación por adelantado y que se ajustan a un plan una vez que se establece, tales modelos no tuvieron éxito al encarar la realidad de los frecuentes cambios ambientales” (SCRUMstudy, 2013, p. 273).

La filosofía de las metodologías ágiles se centra en la satisfacción y colaboración del cliente, en los miembros del equipo, en entregas rápidas e incrementales y en responder al cambio. Las metodologías ágiles se basan en El Manifiesto Ágil, documento que describe las características que estas metodologías deben de tener.

Manifiesto Ágil

El Manifiesto Ágil se elaboró en el 2001, donde expertos se reunieron para discutir las metodologías del desarrollo de software, formaron la *Agile Alliance* y posteriormente con las conclusiones de la reunión Fowler y Highsmith elaboraron el Manifiesto Ágil firmado por la *Agile Alliance*. (SCRUMstudy, 2013, p. 274).

Los 12 principios del Manifiesto son:

1. La máxima prioridad es satisfacer al cliente a través de la entrega temprana y continua de un software de gran calidad.
2. Darle la bienvenida a requisitos cambiantes incluso en una etapa avanzada en el desarrollo.
3. Entregar software de buen funcionamiento con frecuencia, a partir de un par de semanas a un par de meses, con una preferencia por el tiempo más corto.
4. La gente de negocios y los desarrolladores deben trabajar juntos todos los días durante todo el proyecto.
5. Construir proyectos alrededor de individuos, darles el entorno y el apoyo que necesitan y confiar en ellos para hacer el trabajo.
6. El método más eficiente y eficaz de comunicación con y dentro de un equipo de desarrollo es la conversación cara a cara.
7. La medida principal del avance es un software funcional.
8. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y el buen diseño mejora la agilidad.
10. Simplicidad.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto organizados.
12. En intervalos regulares, el equipo reflexiona sobre cómo ser más eficaz, y en base a eso ajusta su comportamiento.

Scrum

El concepto original de Scrum proviene de una jugada de rugby, donde un equipo intenta llegar hasta el final como una unidad, pasando el balón hacia atrás y hacia adelante. Hirotaka Takeuchi y Ikujiro Nonaka definieron una estrategia de desarrollo de producto en la que el equipo de desarrollo trabaja como una unidad para alcanzar un objetivo común.

Scrum es una metodología ágil, Ken Schwaber y Jeff Sutherland elaboraron su aplicabilidad al desarrollo de software en 1995 y posteriormente varios autores han seguido perfeccionando y conceptualizando la metodología Scrum.

“Un proyecto Scrum implica un esfuerzo de colaboración para crear un nuevo producto, servicio o cualquier otro resultado [...], es una metodología de adaptación,

iterativa, rápida, flexible y eficaz, de manera que ofrece un valor significativo de forma rápida en todo el proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. Es compatible en todo tipo de industrias y en cualquier tipo de proyecto, independientemente de su complejidad” (SCRUMstudy, 2013, p. 2).

De manera general Scrum se compone de los siguientes cuatro elementos (SCRUMstudy, 2014): Principios del Scrum, Roles, Artefactos, Reuniones; que si se conocen a profundidad, son suficientes para poder seguir esta metodología:

Principios del Scrum

- **Control del Proceso Empírico.** En Scrum las decisiones deben ser tomadas basadas en la observación y la experimentación, más que en una planificación inicial detallada. Este principio se basa en tres ideas principales: transparencia, inspección y adaptación.
- **Auto-organización.** Scrum cree que los empleados de hoy en día pueden ofrecer mucho más conocimiento que el de sus habilidades técnicas, que son auto-motivados y que buscan adquirir mayor responsabilidad; y debido a esto ellos entregan más valor cuando se organizan por su propia cuenta.
- **Colaboración.** Para Scrum el desarrollo de un producto es un proceso de creación de valor que se comparte y necesita la participación de todos los involucrados, para entregar una mayor cantidad de valor.
- **Priorización Basada en el Valor.** Entregar la mayor cantidad de valor en el menor tiempo posible requiere priorizar y seleccionar aquello que podría ser hecho ahora de lo que debería ser hecho después.
- **Tiempo Asignado.** Scrum trata al tiempo como una de las restricciones más importantes en la gestión de un proyecto y asigna tiempos para cada proceso de acuerdo al ritmo de trabajo de cada uno de los colaboradores, lo cual garantiza que no se use muy poco o demasiado tiempo para una actividad determinada.
- **Desarrollo Iterativo.** Bajo la certeza de que el cliente no siempre estará disponible para definir de manera concreta los requerimientos, el desarrollo iterativo otorga mayor flexibilidad y adaptación a los cambios.

Roles

Los roles en Scrum se dividen en dos categorías llamadas “*Core Roles*” (roles principales) y “*Non-core Roles*”(roles no principales).

Los **Core Roles** son aquellos que obligatoriamente deben existir para ejecutar el proyecto, estos papeles están comprometidos totalmente con el proyecto y son los responsables del éxito de cada iteración o *sprint* y del éxito del proyecto mismo. Estos roles son los siguientes:

- Propietario del Producto (*Product Owner*):
 - Maximiza el valor del negocio para el proyecto.
 - Define la visión del proyecto y el calendario de entregas. Es la voz del cliente.
 - Define los requerimientos del cliente en forma de *Epics* o Historias de Usuario (*User Stories*) y clarifica estos requerimientos a los miembros del equipo Scrum.
 - Prioriza las tareas del producto (*Product Backlog*) de acuerdo al valor del negocio.
 - Define el criterio de aceptación e inspecciona las entregas para validarlas.

- *Scrum Master*:
 - Es un facilitador que asegura que el equipo Scrum tenga un ambiente propicio para completar el éxito del desarrollo.
 - Asegura que los procesos del Scrum sean ejecutados correctamente por el equipo Scrum, incluido el *Product Owner*.
 - Actúa como un líder servicial que ayuda, motiva, enseña y dirige al equipo.
 - Supervisa la Planeación de Entregas y arregla otras reuniones.

- Equipo Scrum (*Scrum Team*):
 - Usualmente es un equipo de 6 a 10 integrantes sin divisiones o sub equipos.
 - Es multifuncional y auto-organizado, tienen completa autonomía durante cada iteración.
 - Comprende los requerimientos de negocio, hace las estimaciones y crea los entregables, la responsabilidad recae sobre todo el equipo.

Los **Non-core Roles** son las funciones que no son necesarias de forma obligatoria para cubrir el proyecto, pueden incluir a miembros de los equipos interesados pero ninguno es parte formal del equipo del proyecto y estas figuras pueden participar con el equipo, sin embargo no son responsables del éxito del proyecto. Pueden considerarse de la siguiente manera:

- Interesados del Proyecto (*Stakeholders*): Es un término colectivo que incluye las siguientes figuras que a menudo interactúan con los *Core Roles* para facilitar el desarrollo del proyecto:
 - Cliente, es aquella persona u organización que adquiere el producto.
 - Usuarios, persona u organización que utilizará directamente el producto o servicio.

- Patrocinador, aquella figura que proporciona los recursos y el apoyo para ejecutar el proyecto. Es el *stakeholder* al que todos deben rendir cuentas al final.
- Proveedores (*Vendors*): en este rol se incluyen a las personas u organizaciones externas que ofrecen productos y servicios que no están dentro de las competencias básicas de la organización del proyecto.
- Cuerpo de Asesoramiento Scrum (*Scrum Guidance Body*): en un rol que no toma decisiones relacionadas con el proyecto pero actúa como consultora en todos los niveles de la ejecución del Scrum.

Artefactos de Scrum

Los artefactos de Scrum son herramientas que forman parte del desarrollo del proyecto y que garantizan que se siga la línea correcta de trabajo para maximizar el valor entregado.

- Definición de la visión del Proyecto: explica las necesidades del negocio, lo que el proyecto está destinado a satisfacer y debe enfocarse en el problema más que en la solución.
- Tareas del producto priorizadas: es una lista de requerimientos priorizados que se pueden convertir en un embarcable o entregable funcional del producto.
- Meta del *Sprint*: Es propuesto por el *product owner* y aceptado por el equipo, y consiste de un objetivo para el *Sprint* actual.
- *Sprint Backlog*: Es una lista de tareas que el *scrum team* se compromete a ejecutar en el *sprint* que está por comenzar, también se incluye cualquier actividad para mitigar algún riesgo.
- Lista de impedimentos (*Impediment Log*): Son los impedimentos y obstáculos encontrados por el equipo y que deben ser debidamente registrados por el *scrum master* en esta lista.
- Incremento del producto (*Product Increment*): es la entrega del embarcable que se genera al finalizar cada *sprint* y que cumple con el Criterio de Aceptación.

Reuniones en Scrum

La metodología de Scrum contempla diferentes reuniones de trabajo para dar seguimiento permanente al desarrollo del proyecto, sin embargo estas reuniones tienen características particulares en cuanto a su duración y frecuencia que permiten

seguir la línea de entrega de valor rápida, a continuación se describen brevemente las más significativas:

- Reunión de Visión del Proyecto (*Project Vision Meeting*): en esta reunión todos los interesados en el proyecto identificarán el contexto del negocio y las necesidades que se desean cubrir en torno a este, se definirán los requerimientos y las expectativas de los interesados (*stakeholders*).
- Reunión de la Planeación de Entrega (*Release Planning Meeting*): El propósito de esta reunión es desarrollar un Plan de Entregas que defina cuando estarán listos para el cliente cada uno de los módulos funcionales del proyecto.
- Reunión de la Planeación del *Sprint* (*Sprint Planning Meeting*): esta reunión debe dar como resultado principal la Lista de Tareas del *Sprint* (*Sprint Backlog*), la planeación y estimación de las tareas debe estar contemplada a cada momento durante la planeación del *sprint*. El tiempo sugerido de la reunión es de 8 horas para ejecutar un *sprint* durante un mes.
- Reunión Diaria (*Daily Standup Meeting*): esta es sin duda la reunión más importante de Scrum ya que permite la comunicación constante de los avances, obstáculos y pendientes durante el *sprint*. Se trata de una reunión diaria de quince minutos, donde cada miembro del equipo Scrum dice que realizó ayer, que realizará hoy y que dificultades tiene para realizar sus tareas en tiempo y forma.
- Revisión del *Sprint* (*Sprint Review Meeting*): en esta reunión el *Scrum Team* presenta las entregas completas del *sprint* que ha terminado, el Propietario del Producto se encargará de aceptar o rechazar los entregables de acuerdo con el criterio de aceptación, se recomienda que esta reunión tenga una duración de cuatro horas para un *sprint* de un mes.
- Retrospectiva del *Sprint* (*Retrospect Sprint Meeting*): los miembros del equipo comparten su experiencia durante el *sprint* que ha finalizado acerca de lo que ha salido bien y lo que no, el objetivo de esto es aprender de la experiencia de todos y mejorar para los siguientes *sprints*. La duración debe ser de 4 horas para un *sprint* de un mes.

Procesos de Scrum

Lo expresado anteriormente, describe en general lo que la metodología Scrum pretende lograr con su implementación y los diferentes medios que emplea, sin embargo no está definido aún el proceso de Scrum y cómo se ejecuta cada una de las fases de esta metodología.

Para poder entender cada uno de los procesos que Scrum indica, es necesario conocer primero el Flujo de Scrum a través del siguiente diagrama:

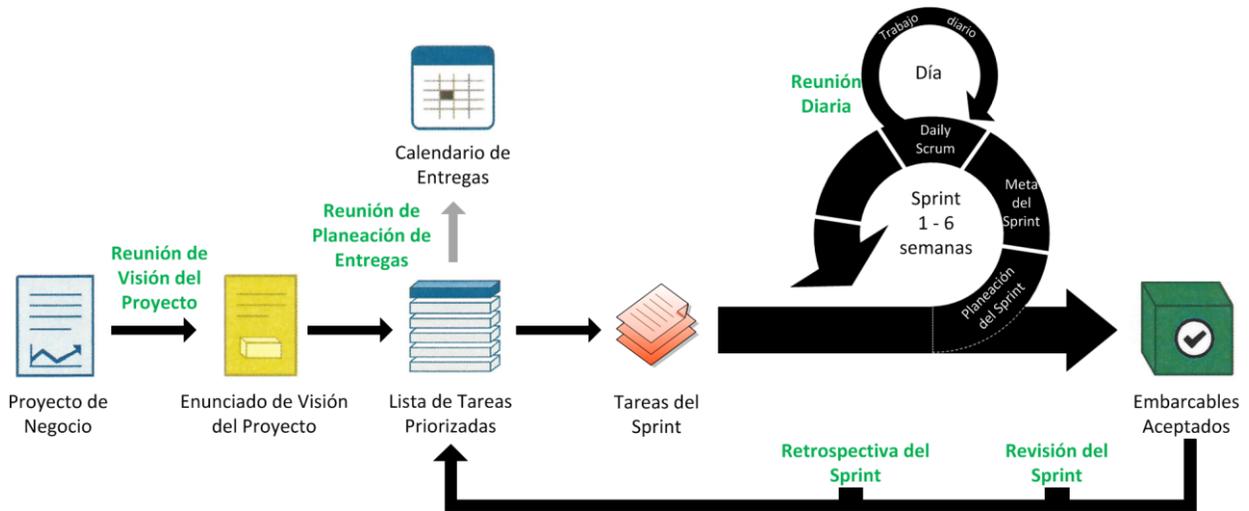


Ilustración 5. Flujo del Scrum. Fuente: SBOK (2013), adaptación propia.

Como se observa, Scrum utiliza una serie de iteraciones llamadas *Sprints*, cada uno de estos consiste en un ciclo de trabajo corto y concentrado, se recomienda que tenga una duración de entre una y seis semanas, y al terminar cada ciclo siempre existirá un embarcable funcional que entregue valor al cliente.

Scrum está dividido en diecinueve procesos agrupados en las siguientes cinco fases de desarrollo para un proyecto completo (SCRUMstudy, 2013, pp. 16 - 19):

- Iniciar
- Planear y estimar
- Implementar
- Revisión y retrospectiva
- Entrega

Los procesos están divididos de esta forma:

Fase Iniciar

1. Crear Visión del Proyecto, a partir de la visión de negocio. Este documento será la guía que conducirá al proyecto para aportar el máximo valor posible.
2. Identificar al *Scrum Master* y a los *stakeholders*, que serán elegidos de acuerdo a diferentes criterios.
3. Crear el equipo Scrum (*Scrum Team*), el Scrum Master propone un equipo de trabajo y el *Product Owner* es quien aprueba la decisión.

4. Desarrollar los *Epics*, basados en la Declaración de Visión del Proyecto que son casos e historias de uso con generalidades del producto, o funciones de alto nivel sin refinar.
5. Crear la Lista Priorizada del Producto, este proceso consiste en priorizar la totalidad de entregables del proyecto que posteriormente se dividirán para cada iteración.
6. Realizar el Plan de Entrega, mismo que definirá la fecha final de lanzamiento del producto.

Fase Planear y Estimar

7. Crear Historias de Usuario, se definirán partiendo de los *Epics* pero con un mayor nivel de detalle con los casos de uso y funcionalidades del producto.
8. Aprobar, Estimar y Asignar Historias de Usuario, a través del *Product Owner* para asegurarse de que entregan valor y embarcables funcionales de acuerdo con las necesidades del proyecto.
9. Crear Tareas, dividiendo todo el proyecto en partes según la cantidad de embarcables planeados.
10. Estimar Tareas, de acuerdo con la experiencia, la velocidad del equipo Scrum y el tiempo límite de entrega, se establecerá el tiempo de entrega de las tareas del proyecto.
11. Crear la Lista de Sprint, para saber qué tareas son las que se harán durante el siguiente *Sprint* y quienes se encargarán de cada una de estas tareas.

Fase Implementar

12. Crear Entregables, es el momento donde se desarrollan las tareas del *sprint*, normalmente se lleva un control sobre un *Scrumboard*, que es un pizarrón con las tareas pendientes, en proceso y hechas para el *sprint* actual.
13. Hacer la Reunión Diaria (*Daily Standup*), para mantener a todo el equipo informado sobre los avances e impedimentos que cada uno tiene.
14. Mantener la Lista Priorizada del Producto, se refiere a actualizar la lista de pendientes constantemente para que el equipo sepa la velocidad con la que se trabaja y se midan los tiempos adecuadamente.

Fase de Revisión y Retrospectiva

15. Convocar un Scrum de Scrums, se refiere a involucrar a diferentes equipos de Scrum que desarrollan un proyecto grande dividido en proyectos más pequeños.
16. Demostrar y Validar el *Sprint*, el equipo Scrum muestra al *Product Owner* el embarcable funcional que se desarrolló durante el *sprint* con el objetivo de que se valide y apruebe el entregable, y en caso de existir cambios se puedan contemplar en el siguiente *sprint*.
17. Retrospectiva del *Sprint*, en este proceso el *Scrum Master* y el *Scrum Team* se reúnen para compartir las lecciones aprendidas a lo largo del *sprint*. Este aprendizaje se documenta para utilizarlo en el desarrollo de los próximos *sprints* e incluso otros proyectos.

Fase de Entrega

18. Enviar Entregables, consiste en hacer llegar los embarcables aceptados hacia los socios o *stakeholders* más relevantes. Se puede escribir un Acuerdo de Entregas Funcionales para documentar la finalización del proyecto.
19. Retrospectiva del Proyecto, es el último proceso del proyecto donde los *stakeholders* y el *Scrum Team* se reúnen para identificar y documentar las lecciones aprendidas durante el proyecto desde su etapa inicial. Usualmente se crea un documento llamado Acuerdo de Mejoras Aplicables (*Agreed Actionable Improvement*) para implementarlas en proyectos futuros.

Evaluación y Mejora del Proceso

Pressman (2010) considera que a pesar de que existen los procesos, no se garantiza que el software se entregue a tiempo, sea de calidad o cumpla con lo que el cliente pidió. Por lo que se han propuesto enfoques para la evaluación y mejora del software para extraer modelos de cómo ejecutar esos procesos para evitar los problemas de crisis del software (p. 31-32).

Como definición del mejoramiento del proceso del Software, se puede mencionar que “el mejoramiento del proceso de software abarca un conjunto de actividades que conducirán a un mejor proceso de software y, en consecuencia, a software de mayor calidad y a su entrega en forma más oportuna” (Pressman, 2010, p. 676).

Se han propuesto enfoques para la evaluación y mejora de los procesos:

- Método de evaluación del estándar *Capability Maturity Model Integration (CMMI)* para el proceso de mejora.
- Evaluación basada en *Capability Maturity Model (CMM)* para la mejora del proceso interno.

- ISO / IEC 15054 estándar que define un conjunto de requerimientos para la evaluación.
- ISO9001:2000 para determinar la calidad de los productos de software.

El *SEI* Instituto de Ingeniería del Software ha realizado trabajos y aportaciones que son referentes, siendo la aportación más significativa los modelos de madurez de las capacidades *CMM* y *CMMI* que tienen como finalidades ofrecer un marco de referencia para mejora de procesos, y ser criterio de evaluación para determinar la madurez, y la fiabilidad de resultados previsibles de una organización de software (Tuya, 2007).

Modelo Integración del Modelo de Madurez de Capacidades (CMMI)

La Integración del Modelo de Madurez de Capacidades (*CMMI*) define las características de proceso si una organización quiere establecer un proceso de software que sea completo (Pressman, 2010, p. 685-686). Es un marco en el que las organizaciones son evaluadas para alcanzar cierto nivel de madurez del proceso y nivel de capacidad. El marco del *CMMI* se presenta en dos formas como modelo continuo y modelo en etapas.

De acuerdo a Sommerville (2005, p. 624) en el modelo continuo no hay una valoración cuantitativa, es una valoración de capacidad para cada proceso. De acuerdo a las necesidades del organismo es posible elegir los procesos que requieren mejorar. Por mencionar algunos ejemplos de procesos que se pueden evaluar, están la gestión de requerimientos, gestión de riesgos, entre otros. De acuerdo al *CMMI* (Software Engineering Institute, p. 36) los niveles de capacidad son cero incompleto, uno realizado, dos administrado y tres es definido.

El *CMMI* en etapas valora las metas y prácticas específicas que se deben alcanzar en cada nivel, siendo cinco los niveles de madurez, los cuales se clasifican según los niveles de capacidad en nivel 0 incompleto, nivel 1 realizado, nivel 2 administrado, nivel 3 definido, nivel 4 administrado cuantitativamente y nivel 5 optimizado. Las metas específicas se desglosan de un proceso, estableciendo las características para que ese proceso sea efectivo, y las prácticas específicas son las que se derivan de esas metas, un ejemplo puede ser que una meta específica sea el desarrollo de un plan de proyecto y como prácticas específicas para lograr esa meta son establecer la calendarización, plan de recursos, plan de riesgos, etc.

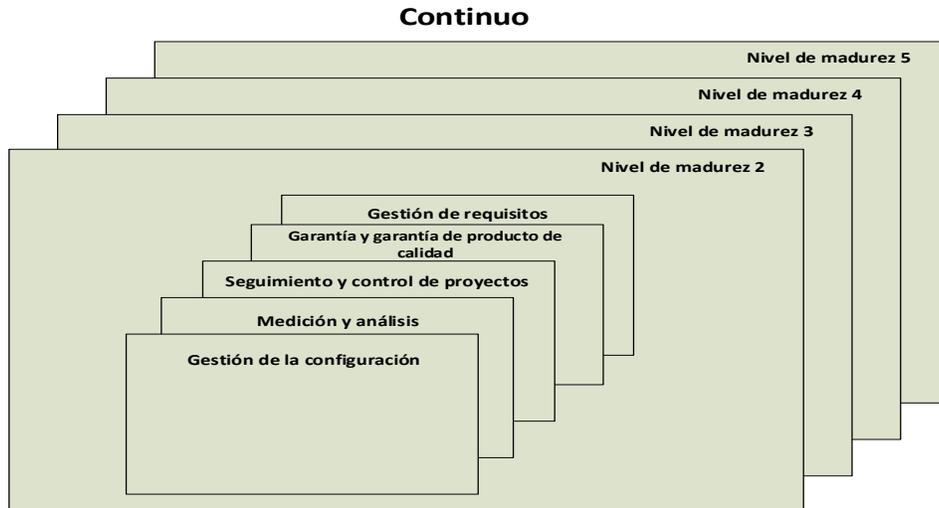


Ilustración 6 Áreas de proceso en las representaciones continua y por etapas. Fuente: adaptada de (Software Engineering Institute, 2010, pág. 31)

La principal diferencia es que el plan en etapas establece niveles de madurez, mientras que el plan continuo son niveles de capacidad. (Pressman, 2010, p. 688). El manual de *CMMI* (Software Engineering Institute, 2010) muestra dos figuras comparativas (ilustración 6).

Los niveles de madurez son: optimización con un enfoque de mejora de proceso continua, el nivel administrado cuantitativamente, el nivel definido encaminado hacia la estandarización del proceso, el nivel administrado orientado a la administración básica del proyecto y el nivel realizado.

Modelo de Madurez de Capacidades del Personal (*People-CMM*)

El modelo de madurez de capacidades del personal “define las áreas prácticas clave para el personal de software: plantilla, comunicación y coordinación, ambiente de trabajo, desempeño administrativo, capacitación, compensación, análisis y desarrollo de competencias, desarrollo profesional, desarrollo de grupo de trabajo y desarrollo de equipo / cultura, entre otros” (Pressman, 2010).

El Personal-*CMM* Modelo de Madurez de Capacidad del Personal valora la calidad de la cultura de la organización y de su población, “para implementar prácticas que mejoren de manera continua la capacidad de la fuerza de trabajo de una organización” (Pressman, 2010, p. 688).

El Personal *CMM* es un conjunto de prácticas de gestión del capital humano que proporciona una ruta para la mejora continua de la capacidad de la fuerza laboral en una organización. Es presentado en etapas, donde en cada nivel progresivo se produce una transformación en la cultura de la organización con prácticas para atraer, desarrollar, organizar, motivar y retener a su fuerza de trabajo (Software Engineering Institute, 2009).

El principal objetivo del Personal *CMM* es mejorar la capacidad de la fuerza laboral, definida como el nivel de conocimientos, habilidades y capacidades de proceso disponibles para la realización de actividades de negocio de una organización. La capacidad de la fuerza laboral indica una organización con:

- Disposición para la realización de sus actividades de negocio críticos, y
- Potencial para beneficiarse de inversiones en mejora de procesos o de tecnología avanzada.

De acuerdo al Personal *CMM* (Software Engineering Institute, 2009, p. 17) son cinco niveles de madurez, un nivel de madurez consiste en prácticas relacionadas sobre un conjunto predefinido de áreas de procesos que mejoren el rendimiento global de la organización. La transformación es evolutiva, para alcanzar un nuevo nivel requiere de capacidades de los niveles anteriores (ilustración 7).



Ilustración 7 Los cinco niveles de madurez del P-CMM. Fuente: adaptada de (Software Engineering Institute, 2009, pág. 36).

En el nivel inicial por lo general se tienen dificultades para retener a las personas con talento. En algunas áreas la organización no ha definido las prácticas de la fuerza laboral o no ha entrenado individuos responsables de realizar las prácticas. Las características comunes son la inconsistencia en la ejecución de prácticas, desplazamiento de la responsabilidad, personal emocionalmente distante. (Software Engineering Institute, 2009, pp. 18-20)

En el nivel dos, nivel administrado, los gerentes deben tomar la responsabilidad personal para el cumplimiento y desarrollo de los que realizan el trabajo, debe prestar atención en la coordinación de los compromisos, provisión de los recursos, desarrollo de habilidades. Se centra en establecer prácticas basadas en las unidades que se ocupan de los problemas inmediatos y preparar a los administradores para implementar prácticas más sofisticadas en los niveles superiores. Algunos de los beneficios de alcanzar el nivel dos es una reducción en la rotación voluntaria, cuando la gente comienza a ver un ambiente de trabajo más racional, mejora la motivación para quedarse con la organización. (Software Engineering Institute, 2009, pp. 20-22)

El principal objetivo del nivel tres, definido, es ayudar a una organización a obtener una ventaja competitiva mediante el desarrollo de distintas competencias que deben ser combinados en su fuerza de trabajo para llevar a cabo sus actividades de negocio. Esas competencias de personal son fundamentales para el plan estratégico de negocios. Es tratar de atar las competencias de la fuerza laboral con los objetivos del negocio actuales y futuros. (Software Engineering Institute, 2009, pp. 23).

Para alcanzar el nivel cuatro, se detectan menos riesgo de trabajo y hay disposición de delegar mayores niveles de autoridad para la gestión de operaciones diarias. Así los directivos son cada vez más libres de gestionar los detalles operativos, y dirigir su atención en cuestiones estratégicas.

En el nivel cinco de optimización, la organización se centra en la mejora continua, se alienta a las personas a hacer mejoras continuas a sus procesos de trabajo personales mediante el análisis de su trabajo y hacer mejoras en los procesos necesarios. Para mejorar la capacidad a nivel de grupo, los procesos de trabajo de cada persona deben integrarse en un procedimiento que funcione para el grupo de trabajo. Las mejoras individuales, deben integrarse en las mejoras en el proceso operativo del grupo. (Software Engineering Institute, 2009, p. 28)

Bases de Datos Relacionales

La mayoría de aplicaciones hoy en día, hacen uso de las bases de datos, ya que a través de ellas pueden acceder, almacenar o manipular datos que les ofrezcan una información significativa. Partiendo de esto, parece que es preciso conocer la diferencia entre dato e información; un dato es un elemento aislado, puede ser un número, carácter o conjunto de caracteres, Setzer define el concepto de dato como “una secuencia de símbolos cuantificados y cuantificables que pueden ser descritos a través de representaciones formales” (Álvarez Caro, 2015, pág. 63), mientras que la información es el procesamiento de los datos, dando un significado de los mismos. Un dato es un elemento aislado pero a través de la interpretación y la puesta en relación con otros datos se puede llegar a obtener información.

Para Pressman (Pressman, 2010, pág. 139) un objeto de datos es una representación de información compuesta que debe ser entendida por el software. Puede ser cualquier cosa que consuma o produzca información, como por ejemplo un informe, una pantalla. Por ejemplo un auto puede considerarse un objeto de dato, el cual tiene ciertas características que son atributos.

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, sus relaciones, semántica y restricciones (Silberschatz, 2006). Existen dos modelos principales de bases de datos: relacional y orientado a objetos. Se revisará en forma general el modelo relacional.

“El modelo relacional de base de datos es llamado así porque se basa en estructuras de almacenamiento de datos llamados relaciones” (Méndez Cruz, s.f.). Las características fundamentales son dos, los datos son percibidos por el usuario como relaciones, y para consultar los datos el usuario cuenta con operadores que generan nuevas relaciones a partir de otras (álgebra relacional). Méndez explica que el modelo relacional se basa en teoría de conjuntos y lógica de predicados.

Pressman (Pressman, 2010) menciona como punto clave que las relaciones indican la manera en la que los objetos se conectan entre sí, y propone un ejemplo entre una persona y un auto, donde la persona posee un auto. Es un ejemplo sencillo y claro de cómo se puede presentar la relación entre objetos.

El diagrama entidad relación conocido como DER, fue propuesto por Peter Chen para diseñar sistemas de bases de datos relacionales (Pressman, 2010, pág. 142). El propósito general del DER es representar objetos de datos y sus relaciones. Los elementos que se quieren modelar son llamados entidades, y se modelan también sus atributos y las interacciones. Considerando el caso de persona y auto, una entidad sería persona y otra entidad sería el auto. Podemos entonces decir que una entidad es cualquier “objeto real o abstracto que existe en la realidad y acerca del cual queremos almacenar información en la base de datos” (De Miguel, 2001).

Arquitectura Modelo, Vista, Controlador (MVC)

El uso de modelos, vistas y controladores para separar el código fuente en un proyecto, es definido por Sommerville (2005, pp. 390-391) como un marco de trabajo o *framework*. Él nos dice también que este marco de trabajo fue propuesto en los 80 como una aproximación al diseño de Interfaces de usuario gráficas (*GUI* por sus siglas en inglés) y que en ese tiempo permitiría presentar los mismos objetos de diferentes formas, con estilos independientes de interacción entre cada presentación; y cuando los datos del objeto son modificados mediante alguna de las presentaciones, el resto de ellas serán actualizadas sin necesidad de ingresar nuevamente los datos para cada una.

Por otra parte, Deitel (Deitel & Deitel, 2004, pp. 699-700) hace referencia a esta técnica como “la arquitectura MVC”, la cual según su punto de vista utiliza varios patrones de diseño y que divide las responsabilidades del sistema en tres partes:

1. El modelo, que mantiene los datos y la lógica del programa;
2. La vista o vistas que proporcionan una presentación visual del modelo y
3. El controlador, que procesa la entrada del usuario y hace modificaciones al modelo.

Menciona también que mediante el uso del controlador, el usuario modifica los datos en el modelo MVC. Después, el MVC informa a la vista acerca de estas modificaciones para que finalmente la misma vista cambie su presentación gráfica para reflejar los cambios en el MVC.

Con las definiciones anteriores, proponemos el siguiente diagrama que ilustra el funcionamiento de la arquitectura:

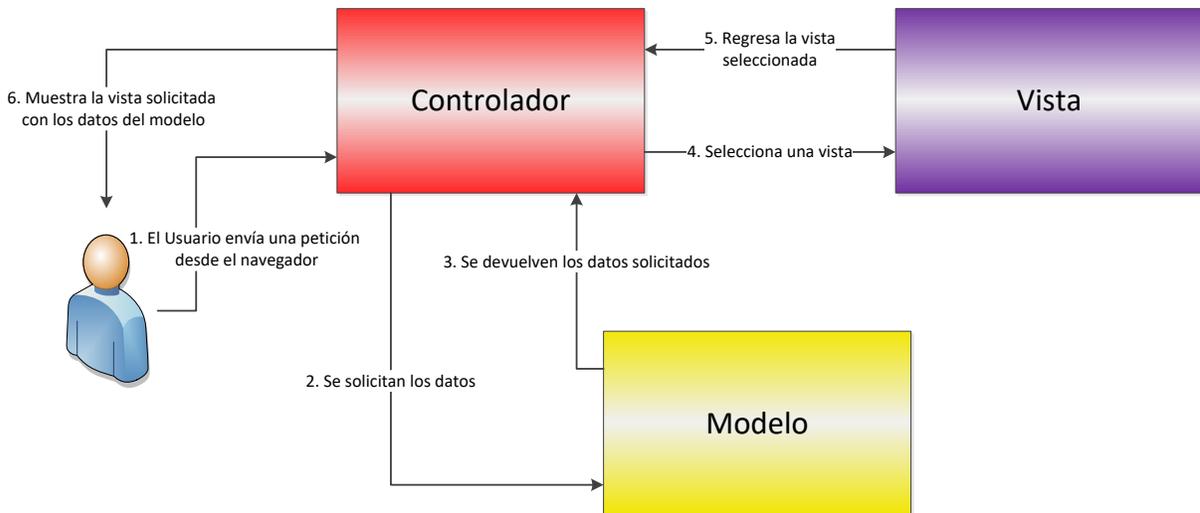


Ilustración 8 Funcionamiento de la arquitectura MVC. Fuente: adaptado de Sommerville (2005).

Es importante mencionar que si bien el diagrama anterior es una adaptación basada en los conceptos definidos por diversos autores, no es un diagrama que ilustre con exactitud la arquitectura MVC ni mucho menos un diagrama que indique una forma única de implementación; pues de acuerdo con otras fuentes, y sobre todo según las necesidades al momento del desarrollo, estas acciones descritas pueden modificarse e incluso pueden incrementar, un ejemplo de otra acción muy utilizada es la interacción directa entre el Modelo y la Vista, esto rompería con lo mostrado en nuestro diagrama, sin embargo la implementación seguiría siendo la de una arquitectura MVC.

Manejador de versiones Subversion (SVN)

Cuando se desarrolla un software, hay un equipo de programadores trabajando sobre un proyecto, en el cual todos deben contar con versiones recientes que incluyan las nuevas funcionalidades o modificaciones que sus compañeros han realizado, esta organización colaborativa puede ser complicada si no se cuenta con un control de versiones sobre la que se trabaja.

“El control de versiones ayuda a manejar información de manera colectiva y que cambia constantemente” (Solsona & Viso, 2007, p. 256).

“El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente, y a otros como documentos, imágenes, sitios web, etc.” (Hernández, 2014, pp. 35).

Subversion es un sistema de control de versiones, su operación involucra un árbol de archivos en un repositorio central, un repositorio es una base de datos central que

contiene todos los archivos cuyas versiones se controlan y sus respectivas historias, éste se encuentra en un servidor de subversion. Un número de clientes se conectan al repositorio y luego leen o escriben en los archivos. (TortoiseSVN team, 2015)

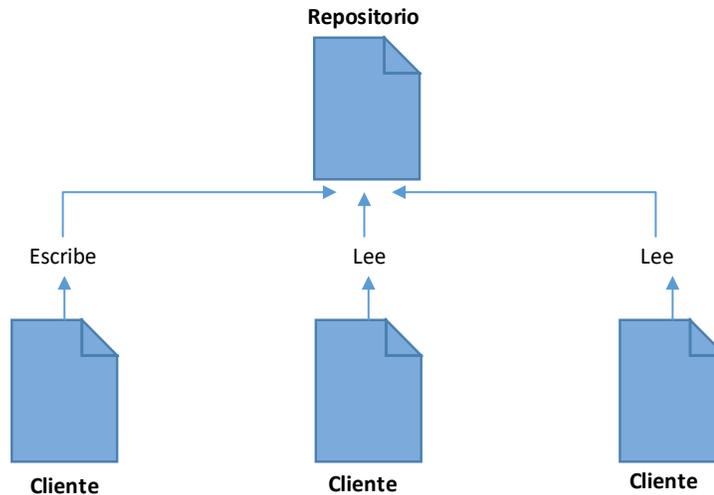


Ilustración 9 Un sistema típico cliente/servidor. Fuente: adaptado de (TortoiseSVN team, 2015)

Así puede haber varias personas en diferentes equipos y lugares físicos, que pueden estar trabajando sobre el mismo proyecto, mejorando de esta manera la colaboración. Dado que se mantiene una historia de las versiones, en el caso de que se haya subido un cambio incorrecto es posible regresar a la versión anterior.

Cada desarrollador tiene una copia de trabajo en su computadora, donde puede tener la última versión que esta almacenada en el repositorio, trabajar sobre la copia local sin perjudicar a nadie, cuando esté seguro de los cambios que ha realizado puede confirmar sus cambios en el repositorio.

Cada individuo debe contar en su máquina con un cliente de subversión, para poder acceder al repositorio central y crear una copia de trabajo, que es un reflejo de los archivos y directorios en el repositorio central. Los usuarios trabajan al mismo tiempo, modificando sus copias privadas y finalmente intercalan sus copias en una versión final nueva.

El ciclo de trabajo de subversión es:

1. Actualizar la copia de trabajo
2. Realizar cambios
3. Revisar cambios
4. Intercalar los cambios de otros en la copia local
5. Comprometer (subir) los cambios al repositorio central

DESARROLLO

En este capítulo de la tesis serán descritos el ciclo de vida del software y la metodología implementada, así como cada uno de los procesos ejecutados durante el desarrollo del sistema planteado.

Investigación de los antecedentes

Al inicio de esta tesis se mencionó la existencia de un sistema web llamado Sistema de Información Automatizada de Servicio Social (SIASS) que fue desarrollado principalmente para concentrar y organizar, en un solo sitio, toda la información generada por las diferentes entidades involucradas con el Servicio Social Universitario de la UNAM.

El sistema fue un éxito durante algunos años ya que resultaba muy sencillo obtener información detallada en poco tiempo y también se reducía la cantidad y el tiempo de espera para los trámites administrativos. Sin embargo con el paso del tiempo este sistema fue presentando deficiencias técnicas y administrativas, las primeras ocasionadas principalmente por el crecimiento de los datos contenidos pero sobre todo por la falta de mantenimiento y actualización de las tecnologías implementadas, y las segundas a causa de los constantes cambios y nuevas necesidades que presentaban los procesos de servicio social.

A pesar de que en la Dirección General de Orientación y Servicios Educativos (DGOSE) se conocían las deficiencias del sistema mencionadas anteriormente, la falta de personal y sobre todo la sobrecarga de trabajo impedía que se ejecutara un plan de mantenimiento y actualización para el sistema que se encontraba en operación, provocando así que el sistema se acercara cada vez más rápido al límite de su capacidad de operación. De esta forma, se propuso la construcción de un nuevo sistema que retomaría las características del SIASS, implementaría las nuevas necesidades del Servicio Social Universitario, se adaptaría a las nuevas tecnologías y que tendría un plan de mantenimiento y actualizaciones para evitar el mal funcionamiento debido al crecimiento de los datos contenidos.

Definición y factibilidad del proyecto

En el momento que la DGOSE decide la creación de un nuevo sistema, la Dirección de Servicio Social se da a la tarea de conformar un equipo que lleve a cabo la ejecución del proyecto. El equipo estaría formado por personal administrativo con amplio conocimiento sobre los procesos del Servicio Social Universitario y por personal técnico y becarios encargados de la administración y desarrollo del proyecto de software. En conjunto se definirían las características generales del proyecto y el plan de acción para conformar una base de referencia para comenzar a trabajar.

El desarrollo de un nuevo sistema era ya un hecho inminente, sin embargo se debían analizar varios aspectos antes de comenzar con un plan de acción. Entre los aspectos más relevantes a analizar se encontraba la factibilidad técnica, es decir, ¿podría desarrollarse este nuevo sistema en tiempo y forma con los recursos humanos, tecnológicos y monetarios con los que se cuenta?

Después de plantear varios escenarios y como respuesta a la pregunta anterior se llegó a la conclusión de que el desarrollo de un nuevo sistema podría ejecutarse en tiempo y forma con el presupuesto otorgado solo de la siguiente manera:

- a) Se solicitarán los servicios de almacenamiento y de red en la DGTIC. De esta forma se ahorra en la compra de infraestructura de hardware y mantenimiento del mismo.
- b) Se contratará un proveedor externo encargado del desarrollo del software. El personal administrativo de la Dirección de Servicio Social cuenta con diversas actividades cotidianas que no pueden desatender en su totalidad y con el ahorro económico en el hardware esta opción es la más viable.
- c) La administración del proyecto de software recae en el equipo de trabajo de la Dirección de Servicio Social.

Después de analizar la factibilidad y tomar las decisiones que se consideraron adecuadas para el desarrollo del proyecto se realizó el siguiente análisis de FODA, que de acuerdo al sitio web de la compañía Matriz FODA (2016), es una herramienta que nos permite analizar y tomar decisiones de acuerdo a los objetivos sobre cualquier situación o ámbito que sea objeto de estudio en algún momento determinado. La sigla FODA se debe al análisis de Fortalezas, Oportunidades, Debilidades y Amenazas.

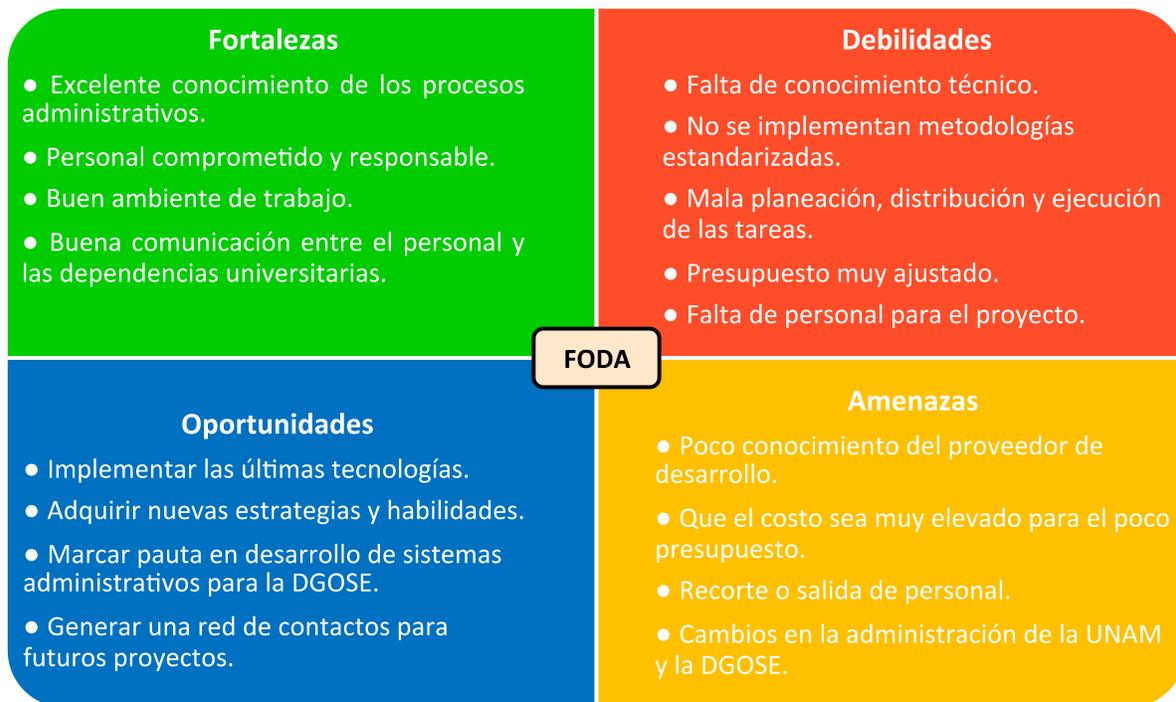


Ilustración 10 Análisis FODA del proyecto. Fuente: adaptación propia.

Inicio del proyecto

El ciclo de vida o modelo de los procesos del software que se ha implementado es el de cascada por ser uno de los más comunes y documentados. Al tratarse de un sistema de software prácticamente construido desde cero, creemos que el modelo de cascada es el más adecuado debido a que, como lo mencionamos en el capítulo anterior, se divide claramente en fases de tal forma que “el cliente” pueda entender fácilmente de qué trata cada una de ellas.

Este ciclo de vida lo hemos complementado con la metodología de desarrollo ágil SCRUM, una metodología iterativa al igual que el modelo de cascada. Las principales ventajas que SCRUM nos ofrece para el desarrollo del sistema son la posibilidad de realizar cambios en un menor tiempo y la entrega de módulos funcionales que aporten valor para el cliente.

De acuerdo a lo que la metodología de SCRUM sugiere, el equipo de desarrollo está conformado de la siguiente manera:

Core Roles

- Propietario del Producto (*Producto Owner*): La Dirección de Servicio Social de la UNAM.

- *Scrum Master*: Coloquialmente son los administradores del proyecto, en este caso nosotros mismos, los desarrolladores de esta tesis Berenice de la Luz y Luis Villagómez.
- Equipo *Scrum (Scrum Team)*: Los desarrolladores encargados de la construcción del sistema.

Non-Core Roles

- Interesados del Proyecto (*Stakeholders*): Personal administrativo de la Dirección de Servicio Social que se encargará de validar los requerimientos, así como de proponer los cambios que considere pertinentes, es la figura de mayor experiencia en los procesos del Servicio Social Universitario.

Las Unidades Responsables de Servicio Social, usuarios finales encargadas de ejecutar los procesos del SSU en cada una de las escuelas y facultades.

El titular de la DGOSE que aportará el recurso económico para el desarrollo y entregará el producto terminado a la rectoría de la UNAM.

- Proveedores (*Vendors*): La DGAE, a través de la Subdirección de Sistemas de Registro Escolar que provee los servicios web para consultar la información de los alumnos de la Universidad.

Y la DGTIC a través del Departamento de Firma Electrónica Avanzada que provee la herramienta necesaria para certificar electrónicamente la conclusión del servicio social.

- Cuerpo de Asesoramiento *Scrum (Scrum Guidance Body)*: la empresa que nos impartió el curso de certificación de SCRUM.

El plan de trabajo se dividió en diferentes actividades priorizadas, y agrupadas en *Sprints* de desarrollo de tal forma que al final, cada *sprint* entregó una parte funcional del sistema.

El desarrollo en *sprints* como lo sugiere la metodología de SCRUM permitió implementar los cambios en el momento adecuado de acuerdo a su nivel de prioridad sin tener que esperar hasta finalizar todo el sistema. Esta técnica permite tener un mejor control sobre los cambios y sobre todo en los tiempos de entrega.

Requerimientos generales

El análisis de los requerimientos es fundamental en la planeación de cualquier proyecto sin importar su naturaleza, en este caso para el desarrollo de un sistema de software no podía ser la excepción.

Como lo sugieren diversos autores de la ingeniería del software, el análisis de requerimientos suele ser una de las primeras etapas en un proyecto, esto le permite al equipo de desarrollo conocer el alcance y las especificaciones detalladas del proyecto que están por realizar.

En nuestro caso el análisis de los requerimientos no fue la primera tarea planeada, como lo sugiere el modelo en cascada, ya que siguiendo la metodología de SCRUM, primero se definieron los roles a desempeñar, el alcance general del proyecto, los *Epics*, la lista priorizada de tareas y los *sprints* del proyecto. Después de lo anterior, nos dimos a la tarea de definir los requerimientos en el primer *sprint*.

En la planeación definimos como un solo *sprint* la elaboración del documento estandarizado de la Especificación de Requerimientos de Software (SRS, por sus siglas en inglés).

Este documento describe detalladamente como funciona cada uno de los módulos del sistema, cuales son las acciones que se ejecutarán en ellos, define también las tecnologías que se utilizarán para la construcción, y todas las reglas de negocio que deberán cumplirse dentro del sistema.

La estructura del documento SRS está basada en el estándar 830-1998 *Recommended Practice for Software Requirements Specifications*, publicado por el Instituto de Ingenieros en Electricidad y Electrónica (IEEE, por sus siglas en inglés).

Acuerdo de trabajo para el desarrollo (Statement of Work, SOW)

En Ingeniería de software y en la administración de proyectos de software es muy común que se establezcan lineamientos y condiciones entre el proveedor y el cliente, estas condiciones se plasman en un documento llamado Acuerdo de Trabajo o *Statement of Work* y generalmente es de carácter legal.

En términos coloquiales podemos verlo como un contrato donde ambas partes, cliente y proveedor, describen y se comprometen a cumplir una serie de condiciones como pueden ser:

- Cuáles serán los entregables y cómo será el producto terminado.
- Plazos o tiempos de entrega.
- Criterios de Aceptación.
- Costos detallados, por entregable, por servicio y costo total del proyecto.

- Multas y recargos por incumplimiento para ambas partes.
- Condiciones de propiedad intelectual.
- Responsabilidades del cliente, del proveedor y mutuas.
- Aspectos legales.
- Otras consideraciones.

Por motivos de confidencialidad, el SOW no aparece como documento anexo ya que contiene información sensible del proyecto. Para efectos de esta tesis únicamente mostramos, con un diagrama de Gantt, la planeación de los tiempos de entrega de cada una de las actividades y el costo final (Trejo Medina, 2011).

Estimación inicial del tiempo de desarrollo

Una vez realizada la toma de requerimientos, se creó la lista de tareas para cada *sprint* de desarrollo y con esto se desarrolló una estimación de los tiempos de entrega.

Para esta planeación nos basamos en la estimación de los tiempos de desarrollo que el proveedor ofreció y en la experiencia sobre la duración de los procesos administrativos propios de la UNAM.

A continuación se muestra la estimación de los tiempos utilizando como herramienta un diagrama de Gantt. Este diagrama es ideal para este proceso ya que muestra la estimación para cada una de las tareas del desarrollo y permite identificar cuales pueden realizarse de manera paralela. Finalmente este diagrama nos indica la fecha final estimada para la entrega.



Nombre	Fecha de inic..	Fecha de fin	Duración
• Sprint 1 Especificación de Requerimientos	13/01/14	28/01/14	12
• Sprint 2 Diseño del Software	29/01/14	18/02/14	15
• Sprint 3 Desarrollo Gestión Institucional	19/02/14	7/03/14	13
• Sprint 4 Desarrollo Administración de Programas	10/03/14	4/04/14	20
• Sprint 5 Desarrollo Gestión Escolar	7/04/14	9/05/14	25
• Sprint 6 Desarrollo Consulta de Programas	12/05/14	30/05/14	15
• Sprint 7 Certificación Electrónica	2/06/14	19/06/14	14
• Sprint 8 Desarrollo Reportes y Estadísticas	2/06/14	20/06/14	15
• Sprint 9 Integración del Sistema	23/06/14	18/07/14	20
• Sprint 10 Capacitación y Pruebas finales	21/07/14	12/08/14	17

Ilustración 11 Gráfica Gantt inicial del proyecto. Fuente: elaboración propia

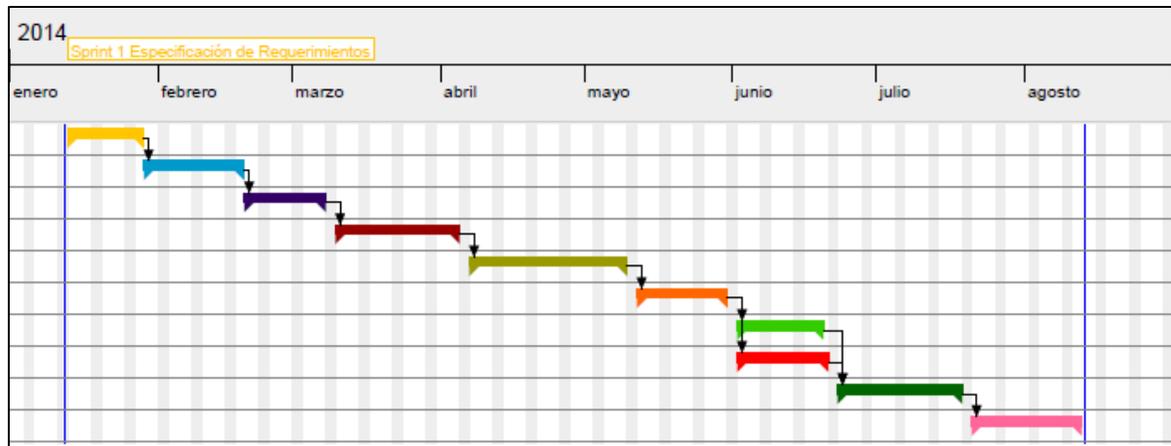


Ilustración 12 Gráfica Gantt inicial del proyecto. Fuente: elaboración propia

Como tal, el desarrollo del proyecto inició el 13 de enero de 2014, con el primer sprint para describir los requerimientos del sistema. Con una duración de 7 meses, el proyecto terminado se entregaría el 12 de agosto de 2014, esta estimación inicial muestra los tiempos acordados entre el cliente y el equipo de desarrollo.

Como se había mencionado anteriormente, cada *sprint* se compone de tareas por realizar, cabe recordar también que estas tareas se deben completar en su totalidad para dar por terminado el *sprint*. Los cambios surgidos durante algún sprint no se efectúan durante el curso del mismo sino al inicio del siguiente el cual debe ser replanteado incluyendo los cambios solicitados como nuevas tareas del *sprint*.

Por lo anterior, el diagrama de Gantt que se presenta está contemplado como el mejor de los casos, ya que de acuerdo a la experiencia, se sabe que todo proyecto involucra cambios durante el desarrollo que no pueden ser planeados en un inicio. Así también y de acuerdo con el cliente, el proyecto podrá ser entregado con un máximo de 30 días naturales de retraso, sin exceder en este periodo de tiempo por la puesta en marcha del sistema programada para el mes de Octubre 2014.

La lista de tareas para los *sprints* mostrados en el diagrama se plantea de la siguiente manera:

Tarea	Fecha de inicio	Fecha de fin	Duración (en días)
Sprint 1 Especificación de Requerimientos	13/01/2014	28/01/2014	12
Descripción de requerimientos	13/01/2014	15/01/2014	3
Elaboración de SRS	16/01/2014	22/01/2014	5
Aceptación de SRS	23/01/2014	24/01/2014	2
Planeación y estimación del proyecto	27/01/2014	28/01/2014	2
Sprint 2 Diseño del Software	29/01/2014	18/02/2014	15
Elaboración SDD	29/01/2014	06/02/2014	7
Aceptación SDD	07/02/2014	07/02/2014	1
Diseño gráfico de la interfaz de usuario	10/02/2014	18/02/2014	7
Diseño de la Base de Datos	10/02/2014	14/02/2014	5
Aceptación Base de Datos	17/02/2014	18/02/2014	2
Sprint 3 Desarrollo Gestión Institucional	19/02/2014	07/03/2014	13
Implementación de la Base de Datos	19/02/2014	21/02/2014	3
Desarrollo e implementación de Acceso por Roles	24/02/2014	25/02/2014	2
Desarrollo gestión institucional	24/02/2014	04/03/2014	7
Pruebas e implementación gestión institucional	05/03/2014	07/03/2014	3
Sprint 4 Desarrollo Administración de Programas	10/03/2014	04/04/2014	20
Desarrollo de Administración de Programas	10/03/2014	28/03/2014	15
Pruebas e implementación de Administración de Programas	31/03/2014	04/04/2014	5
Sprint 5 Desarrollo Gestión Escolar	07/04/2014	09/05/2014	25
Desarrollo comunicación por <i>web services</i>	07/04/2014	11/04/2014	5
Desarrollo Gestión Escolar	07/04/2014	30/04/2014	18
Pruebas e implementación Gestión Escolar	01/05/2014	09/05/2014	7
Sprint 6 Desarrollo Consulta de Programas	12/05/2014	30/05/2014	15
Desarrollo Consulta de Programas	12/05/2014	26/05/2014	11
Pruebas e implementación Consulta de Programas	27/05/2014	30/05/2014	4
Sprint 7 Certificación Electrónica	02/06/2014	19/06/2014	14
Integración de componentes de Firma Electrónica Avanzada	02/06/2014	06/06/2014	5

Desarrollo e Implementación de firmado electrónico	09/06/2014	13/06/2014	5
Pruebas de firmado electrónico	16/06/2014	19/06/2014	4
Sprint 8 Desarrollo Reportes y Estadísticas	02/06/2014	20/06/2014	15
Desarrollo de generación de estadísticas	02/06/2014	06/06/2014	5
Desarrollo de generación de reportes gráficos	09/06/2014	13/06/2014	5
Pruebas e implementación de estadísticas y reportes	16/06/2014	20/06/2014	5
Sprint 9 Integración del Sistema	23/06/2014	18/07/2014	20
Integración total de los módulos	23/06/2014	27/06/2014	5
Pruebas generales del sistema integrado	30/06/2014	02/07/2014	3
Pruebas de seguridad DGTIC	03/07/2014	08/07/2014	4
Documentación de Usuario Final	03/07/2014	08/07/2014	4
Correcciones finales de seguridad	09/07/2014	15/07/2014	5
Instalación en servidor de producción	16/07/2014	18/07/2014	3
Sprint 10 Capacitación y Pruebas finales	21/07/2014	12/08/2014	17
Capacitación a usuarios finales	21/07/2014	01/08/2014	10
Pruebas funcionales y observaciones de usuario final	21/07/2014	08/08/2014	15
Restablecimiento a cero del sistema	11/08/2014	11/08/2014	1
Entrega final	12/08/2014	12/08/2014	1

Tabla 2 Estimación de Sprints. Fuente Elaboración propia

Al ser un proyecto que implementa Scrum como metodología de desarrollo, existen procesos que deben cumplirse antes de siquiera conocer la naturaleza del proyecto que se desarrollará; de acuerdo con la fase inicial de Scrum, la planeación de estos procesos se muestra de la siguiente manera:

Tarea	Fecha de inicio	Fecha de término	Duración (días hábiles)
Inicio: Análisis del Proyecto	18/11/2013	16/12/2013	20
Definición del proyecto de negocio	18/11/2013	22/11/2013	5
Elaboración del enunciado de visión del proyecto	25/11/2013	27/11/13	3
Elaboración de <i>Epics</i> y lista priorizada del producto	28/11/13	16/12/2013	12

Tabla 3 Sprint de planeación y análisis. Fuente: elaboración propia.

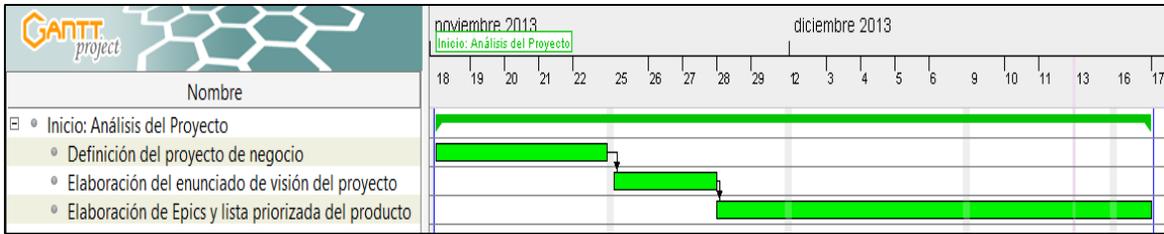


Ilustración 13 Gráfica Gantt inicial del análisis del proyecto. Fuente: elaboración propia

De esta forma, el tiempo inicial total del proyecto desde su concepción será de 9 meses de trabajo iniciando el 18 de noviembre de 2013 para finalizar el 12 de agosto de 2014, a lo largo de este periodo no se generarán avances durante los fines de semana, días festivos y vacaciones administrativas según el calendario de actividades de la UNAM.

Costo estimado inicial del proyecto

Tomando en cuenta la estimación inicial del tiempo de desarrollo, el costo externo de la programación, salarios, becas, compras y otros gastos se realizó la siguiente estimación para el costo inicial del proyecto:

1. Considerando un estimado aproximado del sueldo de los funcionarios administrativos de la Dirección General de Orientación y Servicios Educativos que estuvieron involucrados en el proyecto el costo inicial del proyecto está estimado en \$1,274,500.00 MXN.
2. Si excluimos los sueldos más elevados de estos funcionarios y consideramos únicamente el de los miembros del equipo de desarrollo y administración del proyecto, la estimación inicial en percepciones salariales es de \$552,400.00 MXN.

A pesar de que la duración total del proyecto estipula 9 meses de trabajo, para el cálculo de las estimaciones anteriores se considera el periodo de 10 meses de noviembre 2013 a agosto 2014 en cuanto a sueldos y becas mensuales se refiere.

Estos montos son brutos sin impuestos, y no se considera el uso de instalaciones, equipamiento o costos de seguridad social, se da esta referencia como una métrica inicial de inversión simple.

Diseño de la arquitectura del software

El diseño del software se implementa en este tipo de proyectos para que el equipo de desarrollo tenga claro que es lo que van a construir, es la maqueta de nuestro sistema y muestra un boceto de cada una de las vistas, describe como se efectuará la comunicación con otros sistemas y entre los diferentes componentes del propio sistema.

Complementando lo anterior, el diseño del software describe los diferentes casos de uso del sistema de tal forma que el *Product Owner* y los *Stakeholders* puedan entender cómo será su interacción con el producto terminado y que tipo de acciones podrán ejecutar.

Finalmente, como en todos los sistemas de software, la existencia de documentación que describa detalladamente los procesos permite a los desarrolladores implementar en un futuro estrategias de actualización, optimización y mantenimiento sin que esto se vuelva un caos por no conocer el sistema.

Bajo la misma dinámica de la especificación de los requerimientos del software, el diseño del sistema fue planeado como un solo *sprint*, con la inclusión de nuevos requerimientos, en donde el entregable final sería un documento estandarizado de la Descripción del Diseño del Software (SDD, por sus siglas en inglés).

Este documento SDD está basado en la publicación del estandar 1016-2009 IEEE *Standard for Information Technology – Systems Design – Software Design Descriptions*.

Implementación de la base de datos

Una vez definido el diseño del sistema, se propuso el diagrama relacional para la base de datos de tal forma que este cumplirá con las reglas de negocio y las necesidades del sistema planteadas en la especificación de requerimientos, cubriendo también algunos aspectos tales como consultas para extraer información, respaldos y/o actualizaciones que, por la experiencia, se sabe pudieran presentarse con características muy específicas.

Así, el siguiente diagrama muestra el corazón de la base de datos, que servirá como cimiento de todo el sistema y a su vez almacenará la información más relevante para este sistema de administración del servicio social.

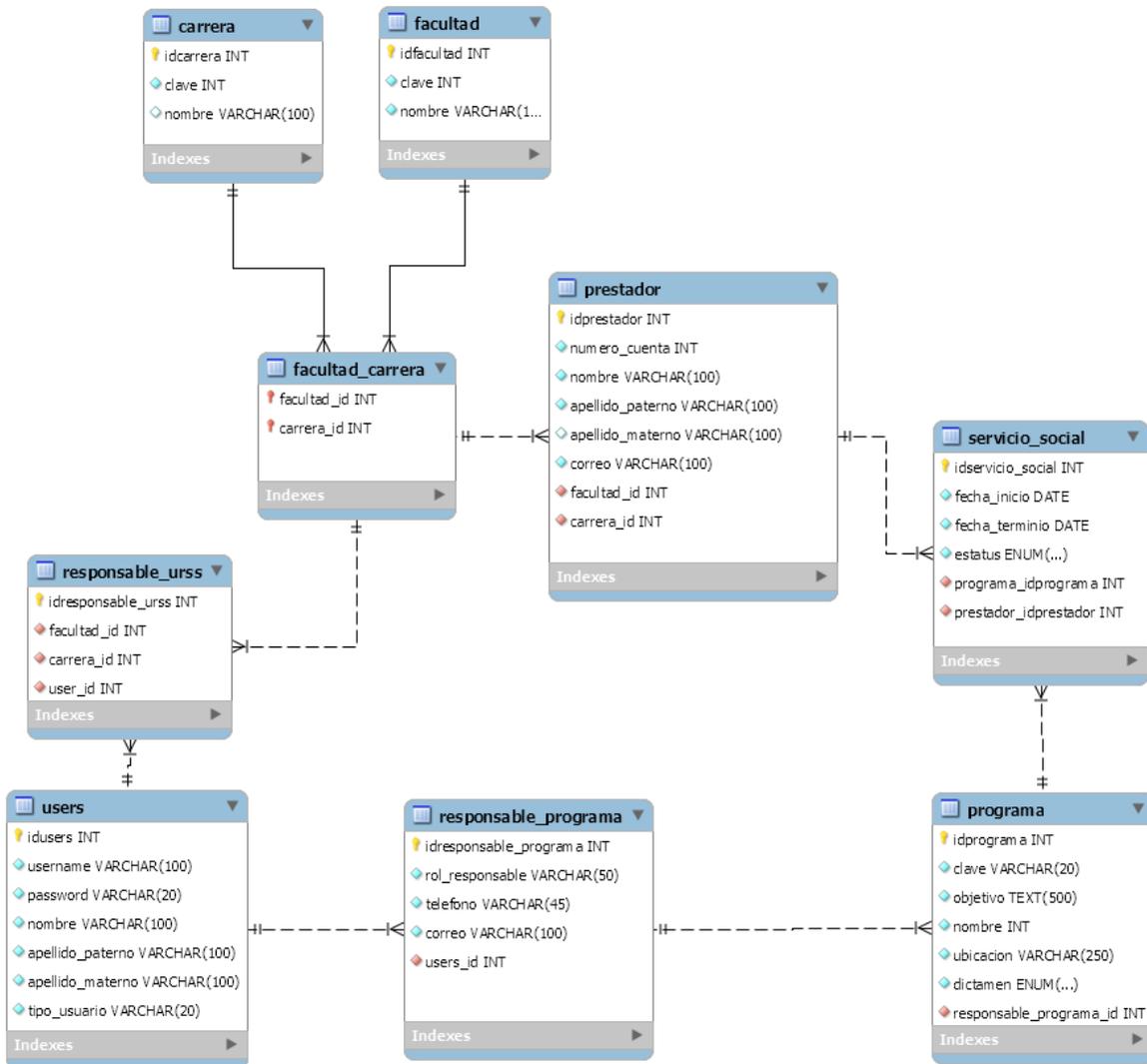


Ilustración 14 Diagrama de diseño de la base de datos. Fuente: elaboración propia

Este diseño se definió de tal forma que la interacción entre las tablas fuera lo más óptima de acuerdo con el tipo de operaciones que se estarán realizando, así como para disminuir en la medida de lo posible la duplicidad de datos que pudieran ser afines a diferentes entidades de esta estructura.

Es importante mencionar que el diagrama mostrado no contiene la totalidad de las tablas y datos que se almacenarán; para fines de esta tesis solo se muestran las tablas principales que permiten cumplir con el proceso de servicio social de la Universidad, reservando en confidencialidad aquellas tablas que terminan por definir la estructura del sistema y los procesos más específicos del mismo servicio social. Así mismo no se muestran las tablas propias del *framework* de desarrollo que, por ejemplo, permiten implementar el acceso basado en roles entre otras acciones necesarias para el correcto funcionamiento del sistema.

Desarrollo del sistema

Cuando nos referimos al desarrollo del proyecto que será realizado por parte de un proveedor externo a la UNAM hablamos de la construcción en código de los módulos y a la documentación de los mismos.

Algunas de las restricciones para la construcción del código fueron que el manejador de base de datos fuera *MySQL* por la extensa documentación existente, el proveedor sugirió el uso del lenguaje de programación *PHP* integrado con el *framework* de desarrollo *Laravel*, pues de esta forma se podría implementar la construcción del sistema y segmentar el código a través del uso de modelos, vistas y controladores.

Pruebas funcionales y de seguridad

Ya que el sistema fue desarrollado y ensamblado en su totalidad se realizaron pruebas de funcionamiento y de verificación de requerimientos, se realizaron las correcciones necesarias para que el sistema ejecute todas las reglas de negocio planteadas en los requerimientos y sobre todo para que no existan errores de programación.

Unos de los criterios de aceptación más importantes para el sistema fue cumplir con los lineamientos de seguridad establecidos por la DGTIC para alojar un sistema en sus servidores; para poder realizar este proceso era necesario cumplir con las pruebas antes mencionadas, con esto obtuvimos “el producto validado”, mismo que ya podía pasar al Departamento de Seguridad de la DGTIC. Una vez aprobadas las pruebas de seguridad, se obtuvo lo que llamamos “el producto terminado”, con esto fue posible proceder a la instalación completa del sistema en el servidor de producción.

Este proceso de pruebas de seguridad se repite n veces hasta que el dictamen ya no arroje más observaciones hacia el sistema, de acuerdo con lo dicho por personal del Departamento de Seguridad, estas pruebas normalmente se ejecutan solo dos veces, de ahí que en la planeación solo se asignaran 4 días hábiles para esta tarea.

RESULTADOS

Durante la elaboración de esta tesis el sistema SIASS se ha puesto en marcha y actualmente se encuentra operando cumpliendo el propósito inicial con el que se justificó el desarrollo del proyecto.

El sistema web se encuentra alojado en la siguiente dirección electrónica: <https://www.siaass.unam.mx/>, donde se muestra la página de inicio para que los alumnos realicen la consulta de los programas de servicio social.

Los beneficios que este sistema trajo para la Dirección de Servicio Social y todas las figuras involucradas en los procesos del Servicio Social Universitario son diversos, pues cada usuario tiene un uso particular del sistema, lo que da lugar al cumplimiento de los objetivos particulares planteados al inicio de la tesis.

En este capítulo se mencionan detalladamente todos los beneficios obtenidos con vistas del producto terminado y cómo se comporta el sistema para cumplir sus propósitos y precisamente ofrecer los beneficios mencionados.

Con la difusión correcta por parte de las Escuelas y Facultades, y de la Dirección de Servicio Social, los alumnos de la Universidad podrán conocer la existencia de este nuevo sistema. De acuerdo al testimonio de algunos alumnos y prestadores, el sistema facilita la búsqueda de programas de servicio social en los que pueden participar, ofreciendo así un abanico amplio de opciones que se adecúan a sus intereses personales y profesionales. De esta forma los alumnos pueden elegir entre programas que ofrecen algún tipo de apoyo, que se ubiquen cerca de su domicilio o escuela solo por mencionar algunas posibilidades.

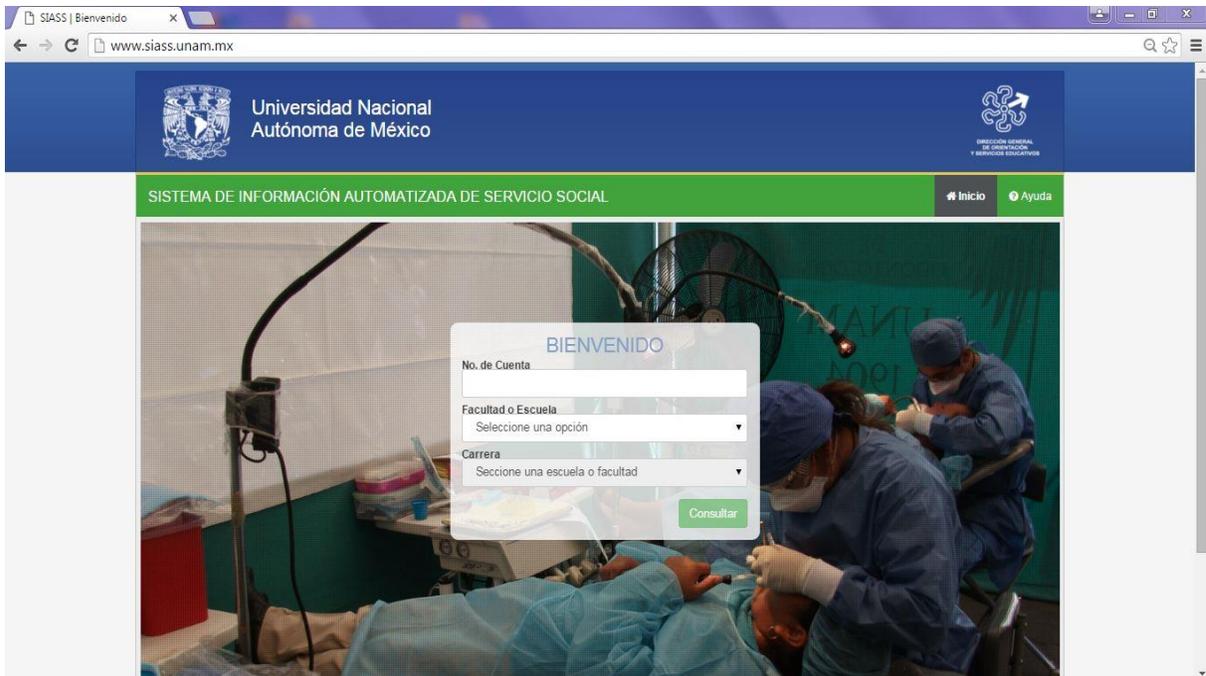


Ilustración 15 Consulta de programas – Inicio. Fuente <http://www.sias.unam.mx>

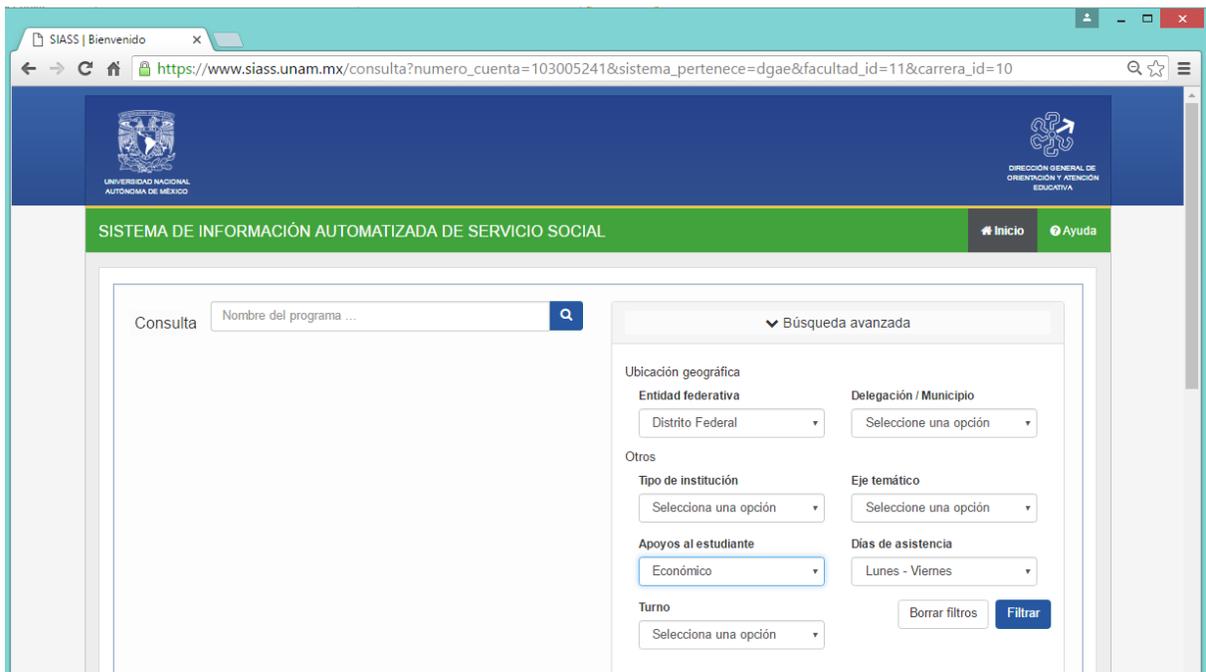


Ilustración 16 Consulta de programas – Filtro de búsqueda avanzada. Fuente <http://www.sias.unam.mx>

En el proceso de entrega final, se hicieron grupos de enfoque en donde mostrábamos la página a estudiantes para validar su utilidad, en dos áreas los que ya habían usado el sistema anterior y quienes no habían usado antes el sistema, en ambos grupos se utilizaron de 8 a 9 personas en una sesión de 30 minutos total, teniendo testimonios positivos en todos los casos y la población estudiantil utilizada como a quienes les presentamos el sistema SIASS nos indicaron de manera directa el beneficio a la población estudiantil.

Una de las compañeras había utilizado el anterior sistema, al primer acercamiento del nuevo SIASS su respuesta fue favorable, comentó que la vista se veía más amigable, tenía una interfaz con un buen diseño, y lo que más le agradó fue que la información del programa del servicio social se presenta de una manera ordenada y que se indica el número de lugares disponibles para el programa, porque le ocurrió que en el anterior sistema no se actualizaba ese dato y algunos programas ya estaban cubiertos en su totalidad, ese fue otro punto a favor que presenta el nuevo SIASS que actualiza ese dato de relevancia para el estudiante, pues en el caso de que no existan vacantes aparece un cero y el estudiante puede seguir buscando otras opciones.

Otras compañeras que aún no realizan su servicio social, externaron que les parecía muy útil y fácil, una de ellas dijo "Me sirvió para conocer y evaluar todos los posibles lugares donde puedo realizar mi servicio social, al momento de ingresar me pareció muy útil que en cada una de las opciones del listado exista contenido más allá de lo básico, algo tan importante como lo son los lugares disponibles para poder laborar en alguna de las varias dependencias que me muestra el sitio. Lo volvería a consultar si fuera necesario. Usaría totalmente dicha página ya que nos facilita la búsqueda de un servicio social, además de que tenemos la posibilidad de poder comparar las distintas opciones de una manera más directa".

Para las Instituciones Receptoras, este nuevo sistema facilita el registro de los usuarios que serán responsables de los mismos programas de servicio social que ofrecerán a los alumnos de la Universidad. El beneficio se encuentra cuando se lleva a cabo el proceso de registro de usuario; para poder realizar este proceso exitosamente, la Institución debe cumplir con una serie de documentos y requisitos administrativos que anteriormente se realizaban en persona o a través de correo electrónico y que ahora se realizan a través del propio sistema reduciendo el tiempo de respuesta; en consecuencia se genera otro beneficio para la Universidad generando un archivo digital de la documentación solicitada a las Instituciones Receptoras.

La sección administrativa del sistema se encuentra alojada en la siguiente dirección electrónica: https://www.siaas.unam.mx/siaas_admin, desde la cual las IR pueden ejecutar sus procesos y así ver reflejados los beneficios mencionados en el párrafo anterior.

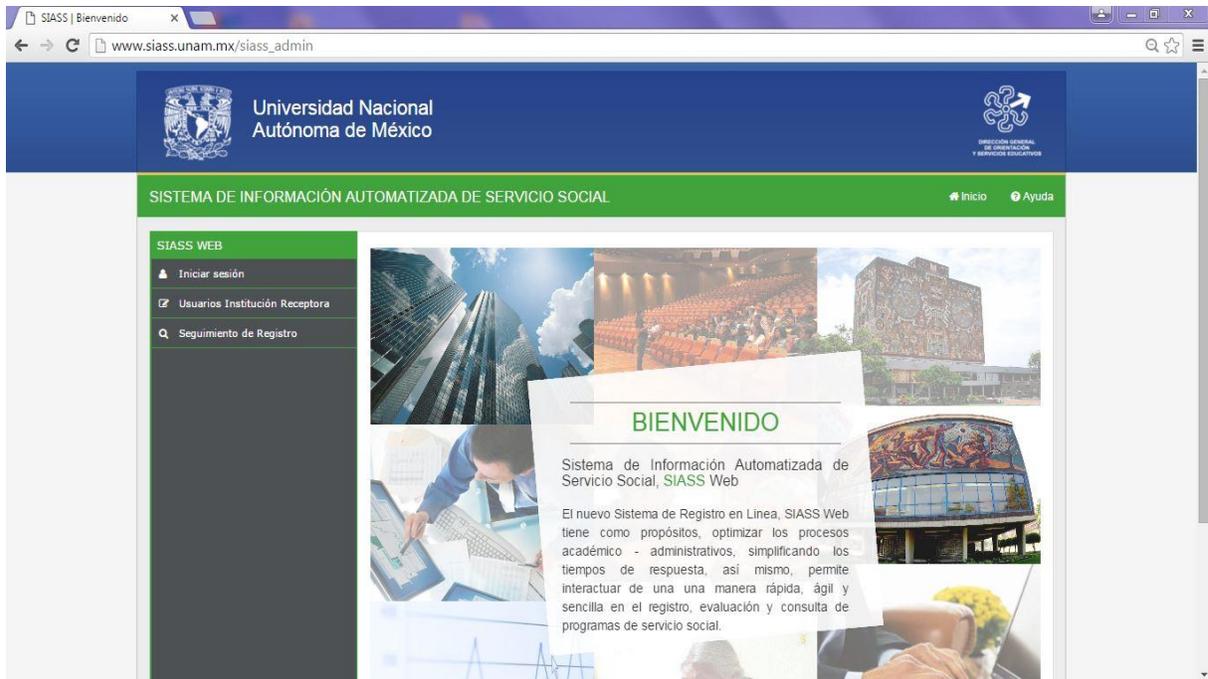


Ilustración 19 SIASS – Inicio. Fuente https://www.siass.unam.mx/siass_admin

Ilustración 20 Registro de Usuario IR (1). Fuente https://www.siass.unam.mx/siass_admin

Información Personal

Nombre *	Apellido paterno *	Apellido materno
<input type="text"/>	<input type="text"/>	<input type="text"/>
Cargo *	Teléfono *	
<input type="text"/>	<input type="text"/>	
	<small>De 8 a 15 números sin espacios</small>	
Correo electrónico *	Confirme su correo electrónico *	
<input type="text"/>	<input type="text"/>	
Rol en la dependencia *		
<input type="radio"/> Coordinador administrativo	<input type="radio"/> Responsable de programa	

Domicilio laboral

Entidad federativa *	Delegación / Municipio *	Colonia / Localidad *
<input type="text" value="Seleccione una opción"/>	<input type="text" value="Seleccione el estado"/>	<input type="text" value="Seleccione el municipio"/>
Calle *	Número exterior	Número interior
<input type="text"/>	<input type="text"/>	<input type="text"/>
Código postal *		
<input type="text"/>		

Ilustración 21 Registro de Usuario IR (2). Fuente https://www.siaas.unam.mx/siaas_admin

Documentación

! Sólo imágenes .JPG y archivos PDF, si el documento contiene más de una página utilice pdf.
El tamaño máximo de archivo es de 2 MB.

Registro de firmas *
Envía el registro de firmas

No se eligió archivo

Código de seguridad *
Por seguridad tienes que escribir el texto de la imagen en la caja de texto.

ERROR: Invalid domain for site key

Ilustración 22 Registro de Usuario IR (3). Fuente https://www.siaas.unam.mx/siaas_admin

En diversas ocasiones se ha mencionado que las IR tienen como propósito registrar los programas de servicio social, para esto cuentan con un formulario sencillo e intuitivo que les permitirá capturar toda la información necesaria para el registro del

programa de servicio social, los beneficios que obtienen con respecto al sistema anterior precisamente son la interfaz y la mejora en la experiencia de uso.

Ilustración 23 Registro de programa de servicio social (1). Fuente https://www.siass.unam.mx/siass_admin

Ilustración 24 Registro de programa de servicio social (2). Fuente https://www.siass.unam.mx/siass_admin

Ubicación geográfica del programa

Ubicación del prestador *

Entidad federativa *

Delegación / Municipio *

Colonia / Localidad *

Calle *

Número exterior

Número interior

Código postal *

¿Las actividades se realizarán dentro de la UNAM? *
 Sí No

Ilustración 25 Registro de programa de servicio social (3). Fuente: https://www.siaas.unam.mx/siaas_admin

Población beneficiaria

Alcance del programa *

Asentamiento *

Población *

Tipo de atención *

Por edad *

Poblaciones marginadas *

Ilustración 26 Registro de programa de servicio social (4). Fuente: https://www.siaas.unam.mx/siaas_admin

Apoyos a Estudiantes

Económico
 Acompañamiento Académico
 Asesoría
 Cursos complementarios
 Atención Médica
 Impresión de Tesis
 Servicios de comedor
 Otros

Asistencia

Días *

Horario *

Ilustración 27 Registro de programa de servicio social (5). Fuente: https://www.siaas.unam.mx/siaas_admin

Otro de los beneficios que se obtuvieron con la implementación de este sistema, derivado del cambio en los lineamientos de servicio social y que se había contemplado desde la toma de requerimientos, tiene que ver con el tiempo de respuesta que la Universidad a través de la DGOSE y las URSS ofrece para la evaluación de los programas de servicio social.

Anualmente se registran alrededor de 6 mil programas de servicio social con 11 carreras en promedio por programa, lo que resulta en aproximadamente 66 mil dictámenes anuales, los cuales anteriormente pasaban a manos de la DGOSE para evaluar las actividades por carrera y dictaminar si corresponden al perfil profesional para así aprobar o rechazar el programa para la carrera en cuestión. Esto generaba una centralización de ese proceso en específico, una sobrecarga de trabajo muy grande para el poco personal administrativo de la DGOSE y sobre todo periodos de hasta 6 meses para evaluar un programa en específico.

Después de la implementación, el nuevo sistema permite que los programas pasen a las Unidades Responsables de Servicio Social de cada facultad para ser evaluados delegando el trabajo que antes recaía sobre 5 ó 6 personas, que no se dedicaban de lleno a eso, y que ahora se encargan de resolver más de 90 usuarios.

De esta forma, se ha logrado un tiempo de respuesta promedio de 6 días hábiles para la evaluación de cada programa. Creemos que estos resultados son realmente significativos ya que se redujo la carga de trabajo de los usuarios de la DGOSE sin afectar la carga de trabajo de los usuarios URSS.

Inicio / Evaluación de programas

Evaluación de programas de servicio social

Filtros

Programa
Nombre ...

Clave
Clave ...

Año
2016

Estado
en proceso

Sistema al que pertenece
DGAE

Escuela o facultad
Seccione una escuela o facultad

Carrera
Seccione una escuela o facultad

Sector
Selección

Institución
Seccione una opción

Dependencia
Seccione una opción

Borrar filtros Filtrar

Mostrando del 1 al 10 de 359

« 1 2 3 4 5 6 7 8 ... 35 36 »

Ilustración 28 Evaluación de programas – Filtro de búsqueda. Fuente: https://www.siasm.unam.mx/siasm_admin

#	Nombre	Clave	Institución	Dependencia	Facultad	Carrera	Estado	Acción
1	Temas Ambientales Y Del Desarrollo Sustentable.	2016-12/176-89	Universidad Nacional Autonoma De Mexico	Programa Universitario De Estrategias Para La Sustentabilidad	FACULTAD DE INGENIERIA	Ingenieria Civil	En proceso	Q Ver detalles
2	Familias En Desarrollo	2016-5/78-1280	Asociaciones Civiles	Asociacion Nacional De Impulso Al Desarrollo Rural Sustentable, A.c.	FACULTAD DE INGENIERIA	Ingenieria Civil	En proceso	Q Ver detalles
3	Programa De Enseñanza Abierta Para Adultos:primaria Y Secundaria Para El Personal Administrativo De Base De La Unam	2016-12/79-1284	Universidad Nacional Autonoma De Mexico	Direccion General De Personal	FACULTAD DE INGENIERIA	Ingenieria Civil	En proceso	Q Ver detalles
4	Programa De Servicio Social De Tutores Pares Para Mejorar El Rendimiento Escolar De La Facultad De Economía	2016-12/59-1472	Universidad Nacional Autonoma De Mexico	Facultad De Economía	FACULTAD DE INGENIERIA	Ingenieria Civil	En proceso	Q Ver detalles
5	Programa De Formación Complementaria, Académica Y Profesional (profocap)	2016-12/25-1467	Universidad Nacional Autonoma De Mexico	Facultad De Estudios Superiores Aragon	FACULTAD DE INGENIERIA	Ingenieria Civil	En proceso	Q Ver detalles
6	Programa Educativo Y Asesoría A La Violencia Escolar (peave)	2016-12/25-1468	Universidad Nacional Autonoma De Mexico	Facultad De Estudios Superiores Aragon	FACULTAD DE INGENIERIA	Ingenieria Civil	En proceso	Q Ver detalles

Ilustración 29 Evaluación de programas – Listado de programas. Fuente: https://www.siasm.unam.mx/siasm_admin

En uno de los objetivos particulares definidos en un inicio, se plantea establecer comunicación con los sistemas de otras dependencias; en este caso específico, la comunicación con DGAE a través de servicios web permite a la DGOSE obtener el avance académico y la información personal de los estudiantes. De la misma forma, DGAE ahora puede conocer el estatus del servicio social de los prestadores y replicar la información del SIASS en sus sistemas para realizar trámites de titulación, mismos que reducen el tiempo de espera debido precisamente a la comunicación en tiempo real entre ambas dependencias.

Siguiendo con los aspectos relacionados a la disminución de cargas de trabajo, los usuarios URSS se ven beneficiados al suprimir la captura de datos para los prestadores que registran, pues como ya se ha comentado, el servicio web devuelve la información más actualizada y confiable de los prestadores de servicio social. Así mismo se eliminan los errores asociados a la captura de datos tales como faltas de ortografía, errores tipográficos entre otros.

Al implementar esta funcionalidad se reduce el tiempo de captura para el registro de prestadores, lo que antes tomaba aproximadamente 25 minutos ahora es posible realizarse en menos de 5 minutos por cada prestador registrado. Si tomamos en

cuenta que anualmente en promedio se registran 20 mil prestadores de servicio social, entonces estamos hablando que se ahorran más de 6 mil seiscientas horas efectivas de trabajo al año.

Inicio / Prestadores de servicio social / Buscar alumno

Ingresa el número de cuenta del alumno

Se usará el Web Service proporcionado por DGAE para buscar al alumno solicitado. Una vez en la vista de detalle del alumno, podrás darlo de alta para asignarle un programa.

Número de cuenta

Sistema al que pertenece

Facultad o Escuela

Carrera

Ilustración 30 Registro de prestadores – Buscar alumno. Fuente: https://www.siass.unam.mx/siass_admin

El módulo de reportes y estadística devuelve cifras concretas y genera gráficas de los números más comunes en materia de servicio social, tales como número de prestadores registrados, programas de servicio social por eje de acción, programas disponibles por carrera, entre otros. Esta transformación de datos en información útil ha beneficiado a los usuarios URSS y DGOSE al momento de presentar resultados periódicos en reportes mensuales trimestrales, anuales etc.

A continuación se muestra un ejemplo de la vista actual del sistema:

Reporte anual del servicio social 2016

Parámetros del reporte ^

Imprimir

Exportar detalle a csv

Programas

Universitarios	2,968
Dependencias seleccionadas	2,968
Externos	2,419
Sector público	1,984
Sector social	435
TOTAL	5,387

Ilustración 31 Generación de reportes – Programas Anuales. Fuente: https://www.siaas.unam.mx/siaas_admin

Reporte de programas registrados (2016)

Parámetros del reporte ^

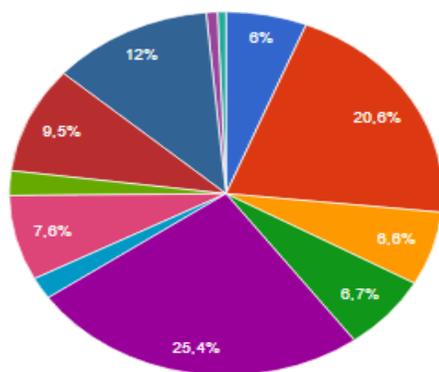
Imprimir

Sector

Ejes de acción

Exportar detalle por eje de acción a csv

Reporte por eje de acción



Eje de acción	Programas
Arte, Cultura y Recreación	325
Ciencia, Tecnología e Innovación	1110
Derechos Humanos, Seguridad Social y Jurídica	354
Desarrollo Social	361
Educación	1368
Equipamiento e Infraestructura Urbana y Rural	105
Medio Ambiente y Desarrollo Sustentable	408
Medios de Comunicación	118
Modernización, Automatización y Fortalecimiento de	511
Salud	648
Seguridad Alimentaria	45
Seguridad Energética	34
TOTAL	5387

Ilustración 32 Generación de reportes – Programas por eje de acción. Fuente: https://www.siaas.unam.mx/siaas_admin

En cuanto a tiempos de desarrollo se refiere, existió un atraso de un tiempo bastante considerable, ya que se habían estimado diez meses desde la gestión del proyecto con fecha de término en agosto de 2014, sin embargo el tiempo total de desarrollo se extendió a 22 meses terminando en septiembre del 2015 a pesar de la planeación propuesta.

Las causas principales de estos atrasos fueron ocasionadas por el proveedor de desarrollo que no entregaba en tiempo los módulos, se extendía de más el tiempo destinado para pruebas y correcciones, las afectaciones debido a la gestión y decisiones directivas que se salían de nuestro alcance, así como los tiempos de respuesta tan largos que se generaron para establecer acuerdos entre las dependencias involucradas en el proyecto, siendo este un punto de riesgo demasiado alto, bajo la metodología de Scrum desde el punto de vista técnico fue sencillo dar seguimiento, pero al tener al *product owner* sin empuje al proveedor externo influyo en el retraso, esto es uno de los puntos fuertes y débiles de Scrum, si el *product owner* está comprometido, el proyecto estaría en funcionamiento rápido, pero al no tener un compromiso en tiempo de entrega se desfasó demasiado. Es necesario aclarar que los tiempos vacacionales y no laborables administrativos influyeron en el retraso, dado que al ser la Universidad dependiente de estas fechas se tenían periodos sin trabajo de varias semanas continuas.

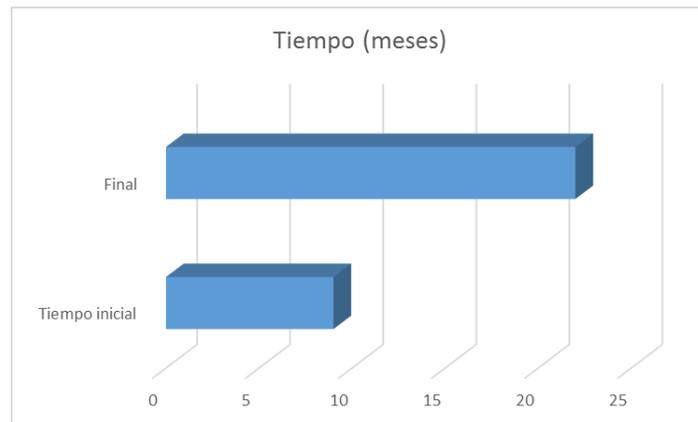


Ilustración 33 Desviación de entrega de proyecto. Fuente: elaboración propia.

Lo anterior repercutió fuertemente en el costo inicial estimado, a continuación se muestra el comparativo entre ese costo inicial y el costo final total del proyecto.

	Con sueldo de funcionarios	Sin sueldo de funcionarios
Costo inicial estimado	\$ 1,274,500.00	\$ 552,400.00
Costo final total	\$ 2,621,900.00	\$ 1,033,280.00

Tabla 4 Estimados de costo. Fuente: elaboración propia.

CONCLUSIONES

El valor aportado a la Universidad con el sistema SIASS es de alto impacto a la comunidad, ya que la actualización tecnológica en la Universidad fue uno de los ejes propuestos por el Rector José Narro en su plan de trabajo 2011-2015. Con el nuevo SIASS, esto se cumple para facilitar y agilizar los procesos del Servicio Social Universitario.

Como se mostró en los resultados, es un sistema intuitivo, de fácil acceso y uso, eso ayuda a que los usuarios puedan utilizar la herramienta al máximo y sacarle el mayor provecho. Esto representa una mejor ejecución del servicio social para todas las partes involucradas.

En cuanto al objetivo general se refiere: implementar un nuevo sistema que contemple los procesos y lineamientos del servicio social y que a su vez optimice los procesos y mejore la funcionalidad; concluimos que se ha cumplido satisfactoriamente, puesto que el SIASS brinda las herramientas necesarias para realizar estos procesos, y nuestro papel de administradores del proyecto fue parte importante para que esto se cumpliera de acuerdo a los lineamientos establecidos en la Especificación del Requerimiento del Software, SRS.

Al inicio de la tesis, se plantearon cuatro objetivos particulares, donde a través de los diferentes módulos del sistema se da lugar al cumplimiento de los mismos.

El primer objetivo particular, mismo que fue cumplido exitosamente es: proporcionar, a los involucrados, una herramienta que facilite y optimice los procesos respectivos para cada uno de ellos. Esto se puede verificar en los resultados anteriormente expuestos, entre los cuales se encuentra el beneficio para las Instituciones Receptoras en lo que al registro como usuarios y de sus programas se refiere.

El tiempo invertido, el envío de muchos correos y el acudir a DGOSE para hacer trámites, ahora son reducidos solo al uso del sistema que facilita los procesos anteriores, ahorrándole a las instituciones receptoras muchas horas. También al finalizar el periodo de vigencia de cada programa, estos podrán ser renovados sin que deban capturarse nuevamente desde cero, en su lugar se podrán realizar solo los cambios que se consideren necesarios si fuera el caso.

Por otro lado, la reducción de tiempos también es a favor de los responsables del servicio social, puesto que ya no tienen que capturar todos los datos del prestador porque el sistema devuelve los campos con la información confiable y actualizada que ofrece DGAE.

El segundo objetivo particular y de mayor impacto es: que los alumnos cuenten con un sitio para consulta eficaz y oportuna de los diferentes programas de servicio social que se ofrecen. Aunque el anterior sistema cubría esta necesidad, con la implementación del nuevo SIASS debía prevalecer y ser mejorada; pudimos

constatar la mejora en la experiencia de uso gracias a los testimonios de compañeros a quienes les presentamos el SIASS y reaccionaron favorablemente por lo práctico e intuitivo que resultó su uso. Sin duda concluimos que se cumplió este objetivo de buena forma ya que el alumno promedio está acostumbrado al uso de las tecnologías de la información y ahora se siente más a gusto con esta funcionalidad que ofrece la ubicación del programa y otras características de interés general.

El tercer objetivo particular se refiere a la creación de una herramienta o funcionalidad en el sistema que permita procesar los datos para obtener información concreta de diferentes tipos. Encontramos el cumplimiento del objetivo con la satisfacción de los usuarios DGOSE y URSS que hacen uso de estas funciones para conocer el número de prestadores registrados, el total de programas para sus carreras y la clasificación por ejes de acción; esta información utilizada comúnmente para la entrega periódica de resultados.

El cuarto y último objetivo se refiere a establecer comunicación sistematizada con otras dependencias de la UNAM, esto se realiza gracias al consumo del servicio web ofrecido por DGAE. Esta tarea implicó diversas actividades, como fueron: la investigación de la tecnología y la comunicación o gestión con otras dependencias, actividad que no necesariamente implica conocimientos técnicos sino habilidades interpersonales, mismas que debemos desarrollar a lo largo del ejercicio profesional y que al ejecutar esta tarea pudimos poner en práctica.

Tomando en cuenta las hipótesis formuladas al inicio de la tesis concluimos lo siguiente:

H1: Si se implementa un sistema de información que utilice los procesos y lineamientos vigentes, utilizando tecnología actualizada se tendrá una mejor administración de la información de los prestadores, instituciones y programas registrados de servicio social en la Universidad.

La primera hipótesis se comprueba y es verdadera; el SIASS responde favorablemente a este enunciado, pues ejecuta todas las reglas de negocio que fueron requeridas y planteadas con base en los lineamientos vigentes del Servicio Social Universitario. Así la DGOSE, quien tiene el control administrativo del sistema, puede usar la información para fines estadísticos, de control y de mejora en la calidad de los procesos del SSU.

H2: Si se integran los datos de los programas registrados del servicio social vigentes en la Universidad, se brindará facilidad de acceso y transparencia a la información para los prestadores.

La segunda hipótesis se comprueba y es verdadera; con nuevas estrategias de difusión y trabajo en conjunto de las escuelas, facultades y la DGOSE, los prestadores del servicio social pueden conocer fácilmente la oferta de programas y las características particulares de los mismos, para, basados en sus gustos y

necesidades, poder tomar la mejor decisión en cuanto a la elección de servicio social se refiere.

En general el sistema aportó muchos beneficios a pesar del costo y tiempo que conllevó todo el proyecto. Una de las deficiencias detectadas, es que aunque se llevó una metodología ágil para la administración del proyecto, las dependencias y funcionarios no están acostumbrados a un ritmo de este tipo, ya que si se hubiera cumplido el plan de desarrollo ágil, se hubiese acabado en un menor tiempo y por lo tanto con menor inversión económica.

Por lo que proponemos que en la Universidad y dependencias se lleve a la práctica las metodologías ágiles, por ejemplo Scrum que puede ser aplicado a muchos campos, no solo al desarrollo del software, esto aportaría más valor y otra perspectiva en la forma de administrar proyectos, sin embargo es importante resaltar que se requiere el compromiso e involucramiento del dueño del proyecto y de la capacitación constante de quienes participan.

Sin duda los conocimientos adquiridos durante la carrera fueron imprescindibles para la elaboración de esta tesis y proyecto, pudimos poner en práctica los conocimientos adquiridos durante nuestra estancia como alumnos. A veces uno como estudiante piensa que solo se llena de teoría y que no puede ser provechoso en algún proyecto, al finalizar este trabajo nos damos cuenta que si no hubiésemos aprendido y entendido la teoría no se habría podido realizar el presente trabajo.

Es en este punto y durante el desarrollo nos dimos cuenta de la importancia de los temas aprendidos en las aulas, y que como ingenieros se nos enseña e inculca el analizar problemas, resolverlos y que las soluciones sean para beneficio de la sociedad. Si bien en algunas materias cursadas no se pudo abarcar el 100% del plan o se vio superficialmente algunos temas, sí ayudaron para darnos un panorama de lo que podría ser útil en la práctica de la ingeniería. Cabe resaltar que al cursar el módulo de salida de ingeniería de software, las materias de éste módulo nos ayudaron para conocer cómo se hace una administración y un desarrollo de software.

Sin embargo aún existen áreas de oportunidad para mejorar la impartición de las materias y el nivel de los profesores. Sabemos que ha entrado un nuevo plan de estudios, que sin duda presenta una mejora pero se debe seguir la línea del trabajo colaborativo e interdisciplinario.

Tomando como referencia la época durante la cual cursamos la carrera, entre las mejoras que hemos observado es que los profesores tengan una constante actualización con las nuevas tendencias en materia de tecnología e ingeniería así como en metodologías y procesos de negocio implementados en el mercado empresarial. Creemos, con el debido respeto, que existen profesores, sobre todo de las materias de ingeniería aplicada, que están desactualizados o separados de lo que ocurre en la industria e imparten temas que ya no son aplicables en el mundo laboral actual, esto representa una desventaja competitiva para los estudiantes y

futuros profesionistas, ya que terminan egresando sin conocer sobre los temas de actualidad y a estas deficiencias se suman la falta de conocimiento en áreas empresariales y de negocio que son indispensables para comprender ese mundo laboral.

En cuanto al módulo de ingeniería de software, algunos profesores sí contaban con experiencia profesional en campo, lo cual no resta merito a los otros, de ninguna manera, pero aquellos que podían compartir esa experiencia nos aportaron un valor agregado importante porque los temas que impartían tenían un respaldo de experiencia real en proyectos empresariales, lo cual validamos durante el desarrollo de esta tesis y que nos apoyó a llevar a buen puerto la misma, aunque con un poco más del tiempo estimado.

Otro punto que se debería mejorar es la vinculación de la carrera de Ingeniería en Computación de la Facultad de Ingeniería con las grandes empresas de tecnología que se desenvuelven en nuestro país, así podrían existir prácticas profesionales o visitas de campo; nosotros tuvimos la oportunidad de hacer una en quinto semestre y nos abrió la expectativa, panorama y visión de la carrera, sin embargo no todos los estudiantes de la carrera lo pueden hacer, pues es muy diferente el acercamiento presencial al lugar donde se lleva a cabo la práctica de la Ingeniería, que el acercamiento a través de la teoría o de las experiencias que cualquier profesor pudiera transmitir vía su cátedra.

En el transcurso de la tesis vimos que es importante el conocimiento técnico, sin embargo en la realización de proyectos también se involucra la parte humana, la que permite relacionarnos con personas que son nuestros directores, supervisores o que están a nuestro cargo, por lo que trabajar en relaciones humanas resulta importante y es algo de lo que carece la facultad, por lo que sugerimos fomentar el trabajo en equipo y talleres de habilidades interpersonales.

También es conveniente integrar a más profesores con experiencia directiva empresarial o de emprendimiento, ya que no es lo mismo tener clase con algún funcionario de la Universidad o de gobierno que con un director de la industria empresarial, existe una diferencia importante entre la visión, estrategia y aporte que se le puede dar a los alumnos por parte de alguien que es empresario, directivo o gerente en actividades de la industria, la forma de pensar nos impulsa a ser líderes, en el otro caso, con el debido respeto, nos deja la enseñanza de sólo ser empleados a largo plazo.

El aprendizaje que obtuvimos con la tesis nos permitió repasar toda la carrera, teniendo como dificultad la expresión escrita de resultados y la comunicación de los procesos, por lo que también creemos firmemente que un ingeniero debe titularse con un proyecto escrito tipo tesis o tesina, ya que consideramos ayuda a obtener una formación integral y es el primer gran problema que resolveremos en nuestra vida profesional fuera de las aulas y sin la total supervisión de un profesor; si bien es válido titularse con un diplomado o extensión de conocimientos, entre otras

modalidades que pueden aportar algo de valor, si vemos una capacidad superior de conocimientos y habilidades en compañeros que optaron por el trabajo escrito.

En nuestra experiencia nos hemos topado con ingenieros sin tesis, incluso algunos experimentados, que les cuesta entablar de forma adecuada comunicación tanto oral como escrita, al redactar informes, manuales o reseñas. A veces carecen de forma y orden para expresar sus ideas y lo más común es que no logran ser capaces de comunicarse apropiadamente en un lenguaje común o coloquial, de tal forma que cualquier persona, sin conocimientos técnicos, pueda entender lo que se quiere transmitir.

Sabemos que no somos los más experimentados en temas de expresión y comunicación, pero a nuestro parecer estas habilidades, que pueden ser desarrollables, deberían estar presentes en todos los ingenieros ya que son casi naturales, también creemos que si hemos logrado detectar estas deficiencias en nuestros colegas es porque algo estamos haciendo diferente y que probablemente esto nos llevará a tener éxito en la vida profesional.

Para finalizar concluimos con la idea de que todos los ingenieros deben ser líderes en sus organizaciones y equipos de trabajo porque nuestra capacidad de abstracción para resolver problemas siempre será una ventaja competitiva sobre cualquier otro profesionista; sobre todo si se trata de perfiles que no son afines a la ciencia y tecnología.

REFERENCIAS

- A guide to the Project Management Body of knowledge.* (2000). Recuperado el 14 de enero de 2016, de <http://www.cs.bilkent.edu.tr/~cagatay/cs413/PMBOK.pdf>
- Álvarez Caro, M. (2015). *Derecho de las nuevas tecnologías*. Madrid: Reus, S.A.
- Camarena, J. M. (2000). La naturaleza del trabajo. Ser y hacer. *Surgir. Las ciencias, las artes, la fe.*(4), 5-2. Obtenido de <http://juanmanuelsilvacamarena.com/D%20Textos%20JMSC%2076-100/89.%20Ser%20y%20hacer/Ser%20y%20hacer.html>
- Consejo Académico del área de las Ciencias Físico Matemáticas y de las Ingenierías. (s.f.). *Programa de Estudio Álgebra*. Recuperado el 9 de Octubre de 2015, de <http://www.ingenieria.unam.mx/paginas/Carreras/planes2006/Computacion/01/algebra.pdf>
- Consejo Académico del rea de las Ciencias Físico Matemáticas y de las Ingenierías. (s.f.). *Programa de Estudio Ética Profesional*. Recuperado el 12 de Octubre de 2015, de http://www.ingenieria.unam.mx/paginas/Carreras/planes2006/Computacion/04/etica_profesional.pdfhttp://www.ingenieria.unam.mx/paginas/Carreras/planes2006/Computacion/04/etica_profesional.pdf
- Consejo Académico del Área de las Ciencias Físico Matemáticas y de las Ingenierías. (s.f.). *Programa de Estudio Ingeniería de Software*. Recuperado el 9 de Octubre de 2015, de http://www.ingenieria.unam.mx/paginas/Carreras/planes2006/Computacion/05/ingenieria_de_software.pdf
- De Miguel, A. (2001). *Diseño de bases de datos relacionales*. Madrid: Alfaomega-Roma.
- Definición de Abstracción.* (s.f.). Recuperado el 09 de Octubre de 2015, de Definición.De: <http://definicion.de/abstraccion/#ixzz3o0FoA7Ad>
- Deitel, H. M., & Deitel, P. J. (2004). *Como Programar en JAVA* (Quinta ed.). México: Pearson Educación.
- Dirección General de Orientación Y Servicios Educativos (DGOSE). Universidad Nacional Autónoma de México (UNAM). (2015, Marzo 12). *Servicio Social - Definición*. Recuperado el 12 de Octubre de 2015, de <http://www.dgoserver.unam.mx/portaldgose/servicio-social/htmls/ss-universitario/ssu-definicion.html>
- Dirección General de Orientación Y Servicios Educativos (DGOSE). Universidad Nacional Autónoma de México (UNAM). (2015, Marzo 12). *Servicio social - Marco Legal*. Recuperado el 25 de Octubre de 2015, de <http://www.dgoserver.unam.mx/portaldgose/servicio-social/htmls/ss-universitario/ssu-marco-legal.html>
- Facultad de Ingeniería, 7.0.* (s.f.). Obtenido de Servicio Social: <http://www.ingenieria.unam.mx/~serviciosocial/>
- Facultad de Ingeniería. Universidad Nacional Autónoma de México (UNAM). (15 de Octubre de 2006). *Facultad de Ingeniería*. Recuperado el 20 de Octubre de 2015, de http://www.ingenieria.unam.mx/paginas/Carreras/planes2006/ingComputo_Plan.htm

- Hernández Rodríguez, J. I. (2014). *Análisis y desarrollo web*. Obtenido de <https://books.google.com.mx/books?id=nYDVBQAAQBAJ&lpg=PP1&hl=es&pg=PP1#v=onepage&q&f=false>
- IEEE Computer Society. (2005). Software Engineering. En *The Development Process* (Tercera ed., Vol. 1). New Jersey, Estados Unidos: A Jhon wiley & Sons, Inc. Publications.
- IEEE Computer Society. (2009). *Std. 1016-2009*. Recuperado el Junio de 2011, de IEEE Xplore.
- IEEE Computer Society. (2014). SWEBOK V.3. En *Guide to the Software engineering Body of Knowledge*.
- Jacobson, I. (1992). *Object-Oriented Software Engineering*. Addison-Wesley.
- Jacobson, I. (enero de 2002). A Resounding Yes to Agile Processes - But Also More. *Cutter IT Journal*, 15(1), 18-24.
- Matriz FODA. (02 de Septiembre de 2016). *¿Qué es la matriz FODA?* Recuperado de <http://www.matrizfoda.com/dafo/>
- Méndez Cruz, C. F. (s.f.). Recuperado el 18 de enero de 2016, de <http://fcasua.contad.unam.mx/apuntes/interiores/docs/2005/informatica/3/1365.pdf>
- Oasis. (s.f.). Recuperado el 6 de enero de 2016, de Advancing open standards for the information society: <https://www.oasis-open.org/committees/download.php/24846/Example-SoftwareDesignDocument-LegalXMLUtility.pdf>
- Pressman, R. S. (2010). *Ingeniería del software, Un enfoque práctico* (Séptima ed.). McGrawHill.
- Rational software. (1998). *IBM Rational*. Recuperado el 27 de octubre de 2015, de Rational Unified Process, Best practices for software development teams: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- Real Academia Española. (2014). *Diccionario de la lengua española*. Recuperado el 2015, de <http://dle.rae.es/index.html>
- SCRUMstudy. (2013). *Una guía para el Conocimiento de SCRUM (Guía SBOK)*.
- SCRUMstudy. (2014). *Scrum On A Page v4*. Obtenido de SCRUMstudy.com
- Silberschatz, A. (2006). *Fundamentos de bases de datos* (Quinta ed.). Madrid, España: McGraw Hill.
- Software Engineering Institute. (s.f.). Recuperado el 11 de enero de 2016, de <http://www.sei.cmu.edu/reports/10tr033.pdf>
- Software Engineering Institute. (2009, julio). *CMMI for people*. Recuperado el 18 de enero de 2016, de http://resources.sei.cmu.edu/asset_files/TechnicalReport/2009_005_001_15095.pdf
- Software Engineering Institute. (2010, Noviembre). *CMMI for Development*. Recuperado el 13 de enero de 2016, de <http://www.sei.cmu.edu/reports/10tr033.pdf>
- Solsona, F., & Viso, E. (2007). *Manual de supervivencia en Linux* (Primera ed.). México: Facultad de Ciencias (UNAM).
- Sommerville, I. (2005). *Ingeniería del Software* (Séptima ed.). Madrid: Pearson Educación.

- Sommerville, I. (2011). *Software Engineering* (novena ed.). USA: Pearson Education.
- Swokowski, E., & Cole, J. (2006). *Álgebra y trigonometría con geometría analítica* (11 ed.). México: Thomson.
- Tamayo y Tamayo, M. (1999). *El Proyecto de Investigación* (Tercera ed.). Cali: Arfo Editores LTDA.
- TortoiseSVN team. (2015, agosto 20). *TortoiseSVN*. Recuperado el 18 de enero de 2016, de https://tortoisesvn.net/docs/release/TortoiseSVN_es/tsvn-basics.htm
- Trejo Medina, D. (2011). *Apuntes de Ingeniería de Software*. México, CDMX, México: DanTM.
- Tuya Javier, R. R. (2007). *Técnicas cuantitativas para la gestión en la ingeniería de software*. España: Netbiblio.
- Universidad Nacional Autónoma de México (UNAM). (1985). *Reglamento General de Servicio Social de la Universidad Nacional Autónoma de México*. Ciudad Universitaria, D.F.

ANEXOS

Especificación de Requerimientos de Software (*Software Requirements Specification*)

1. Introducción

1.1. Propósito

Siguiendo las prácticas recomendadas en el estándar 830 “*Recommended Practice for Software Requirements Specifications*” del Instituto de Ingenieros en Electricidad y Electrónica (IEEE por sus siglas en inglés), este documento tiene como propósito dar a conocer a los involucrados del proyecto, el conjunto de requerimientos, sobre los cuales se construirá el sistema, también presenta su funcionalidad general.

1.2. Alcance

El Sistema de Información Automatizada de Servicio Social (SIASS) es un sistema de software orientado a cumplir los lineamientos y necesidades del Servicio Social Universitario, que cumplirá con el propósito de optimización de los procesos actuales.

El sistema ofrece herramientas de administración para Instituciones Receptoras, Unidades Responsables de Servicio Social, personal administrativo de la Dirección de Servicio Social.

Complementariamente ofrece el servicio de consulta de programas deservicio social para los alumnos.

1.3. Definiciones, acrónimos y abreviaturas

Término	Definición
CURP	Clave Única de Registro de Población.
CRUD	Funciones básicas de una base de datos. <i>Create, Read, Update and Delete</i> .
DGAE	Dirección General de Administración Escolar.
DGOSE	Dirección General de Orientación y Servicios Educativos.
DGTIC	Dirección General de Cómputo y Tecnologías de Información y Comunicación.
IR	Instituciones Receptoras.
MVC	Estructura del código Modelo-Vista-Controlador.
RAM	Memoria de Acceso Aleatorio por sus siglas en inglés (<i>Random Access Memory</i>).

RGSS	Reglamento General de Servicio social.
SIASS	Sistema de Información Automatizada de Servicio Social.
SRS	Especificación de Requerimientos de Software (<i>Software Requirement Specification</i>).
SSU	Servicio Social Universitario.
UNAM	Universidad Nacional Autónoma de México.
URSS	Unidades Responsables de Servicio Social.

1.4. Referencias

Este documento está elaborado basándose en el estándar 830-1998 *Recommended Practice for Software Requirements Specification* publicado por el Instituto de Ingenieros en Electricidad y Electrónica (IEEE por sus siglas en inglés). Referenciado del libro de la IEEE *Software Engineering* (IEEE Computer Society, 1998, pp. 168-187).

IEEE Computer Society. (2005). Software Engineering. In *The Development Process* (Tercera ed., Vol. 1). New Jersey, Estados Unidos: A Jhon wiley & Sons, Inc. Publications.

IEEE Computer Society. (1998). IEEE Standard 830-1998 Recommended Practice for Software Requirements Specification. New York: IEEE.

1.5. Resumen

El documento está dividido en 3 partes:

- La sección 1 se enfoca en la explicación, objetivos, metas y descripción del documento de requerimientos.
- La sección 2 contiene la descripción general del sistema y las reglas de negocio orientadas al cliente y a los usuarios finales.
- La sección 3 define los requisitos específicos del sistema. Se emplean un lenguaje más técnico orientado a los desarrolladores y a los administradores del proyecto.

2. Descripción General

2.1. Perspectiva del producto

Aunque se contaba con un sistema web para la administración del servicio social, éste no cubría el total de necesidades del proceso del SSU y debido a la falta de documentación del sistema resultó imposible implementar un plan de actualizaciones sobre el mismo; además de lo anterior, la Universidad se encuentra en un proceso de replanteamiento de sus procesos más significativos, con lo que se busca reducir el uso de papel y aumentar el uso de las tecnologías de la información.

Así es como el SIASS será un nuevo desarrollo independiente del anterior. Tendrá comunicación bidireccional con otros sistemas, implementando el uso de servicios web con otras dependencias, específicamente la Dirección General de Administración Escolar (DGAE) para el envío y obtención de información que complemente nuestro sistema.

2.1.1. Sistema de interfaces

El sistema de interfaces deberá corresponder con las reglas de negocio propias del sistema, mostrando así diferentes vistas y opciones de menú de acuerdo con el tipo de usuario que inicie sesión en el sitio, los roles y tipos de usuario se definirán en otra sección de este documento.

2.1.2. Interfaz de usuario

El sistema deberá contener una interfaz de usuario sencilla y amigable para el usuario final, institucional y definida por el cliente, es este caso la DGOSE.

La implementación de la interfaz gráfica del sistema deberá quedar tal cual como se solicita y será de tipo responsivo, lo que significa que se adaptará al tamaño de la pantalla del dispositivo o de la ventana donde se ejecute el sistema.

2.1.3. Interfaz de hardware

Al ejecutarse sobre un navegador web, el sistema no requiere interfaces de hardware adicionales a las del dispositivo que contiene el navegador ya mencionado.

2.1.4. Interfaz de software

El sistema deberá ser ejecutado sobre un navegador web y con acceso a internet, ser compatible y totalmente funcional con Google Chrome, Mozilla Firefox, Safari e Internet Explorer 10 en adelante.

Las interfaces de software del sistema estarán definidas por los diferentes botones de la página y las acciones asociadas a cada uno de ellos. El uso de los diferentes botones será mediante las señales enviadas a través del puntero de la computadora.

Adicionalmente el sistema deberá contener un complemento de Adobe Reader para que a través de una ventana emergente propia del sistema se visualicen los documentos PDF almacenados.

2.1.5. Comunicación de interfaz

La comunicación con los diferentes componentes externos y la información propia del sistema será a través del protocolo HTTPS y transportando la información cifrada.

Complementando este requerimiento, el sistema deberá cumplir con las especificaciones que la DGTIC establece para el alojamiento de sistemas en sus servidores. Estas especificaciones serán notificadas a la DGOSE a través de un informe de la revisión de seguridad que se ejecutará cuando el sistema cuente con el cien por ciento de la funcionalidad inicial.

2.1.6. Limitaciones de memoria

La DGOSE, de acuerdo a sus necesidades de almacenamiento anual, concurrencia y velocidad de procesamiento, solicitará alojamiento compartido (virtual host) del sistema en los servidores de la DGTIC; lo establecido en dicha solicitud establece el límite superior de capacidades, por lo que el sistema deberá ser diseñado y construido para operar en condiciones por debajo de ese límite.

2.1.7. Operaciones

Las operaciones que serán ejecutadas en el sistema estarán definidas de acuerdo a lo especificado en el acceso basado en roles así como por procesos internos que no dependerán de la interacción directa de los usuarios.

Las operaciones interactivas serán aquellas en donde los usuarios a través del inicio de sesión ejecuten acciones del CRUD del sistema, tales como registro de prestadores o evaluación de programas de servicio social. Así mismo se incluye la consulta de dichos programas ejecutada por los estudiantes que no requieren autenticación en el sistema.

Las operaciones propias del sistema que serán transparentes para los usuarios finales y contemplarán la generación de dictámenes para cada programa, la actualización automática del catálogo de programas aprobados, la inserción de datos en las tablas asociadas a los modelos donde se ejecutan las operaciones interactivas y la emisión automática de notificaciones, por mencionar algunas.

El sistema y sus componentes (esquema, datos, *framework*, dependencias, etc.) contarán con una serie de operaciones de respaldo y recuperación que será propuesta por los desarrolladores del sistema y aprobado por DGOSE en conjunto con DGTIC.

2.1.8. Adaptación de requerimientos

Todas las características y casos de uso particulares que se planteen para el sistema y que requieran modificaciones sobre el plan y diseño iniciales serán evaluados por la DGOSE quien determinará si se implementan o no, así como el plan de acción según corresponda para cada caso.

2.2. Funciones del producto

El producto terminado tendrá que cubrir las siguientes funciones generales:

- a) Administración de usuarios involucrados (Altas, bajas, Cambios) como prestadores de servicio social, instituciones receptoras de servicio social, responsables de servicio social y personal de la DGOSE.
- b) Administración, recepción y emisión de los trámites de servicio social. (Inscripción de programas de servicio social de las Instituciones Receptoras, seguimiento, y emisión de constancias de inicio y término).
- c) Consulta eficaz y oportuna de los diferentes programas de servicio social.
- d) Generación de reportes, a través del procesamiento de los datos del sistema de información de servicio social para traducirlos en información estadística.

2.3. Características del usuario

Se mencionan los usuarios que actuarán en el sistema:

- a) Prestadores de servicio social
- b) Personal Administrativo de la Dirección General de Orientación y Servicios Educativos (DGOSE).
- c) Institución Receptora (IR).
- d) Unidades Responsables de Servicio Social (URSS).

En general los usuarios deben contar con un nivel de experiencia básico en cuanto al manejo de computadora, y un conocimiento mínimo respecto a la navegación en internet y operación de sistemas web. Este nuevo sistema está orientado a usuarios finales que no requieren de experiencia técnica adicional a la ya mencionada.

Para una operación más fluida, los usuarios de Instituciones Receptoras, Responsables de Servicio Social y de DGOSE deberán conocer detalladamente los términos de cada uno de los procesos que se llevan a cabo en el Servicio Social Universitario y que se reflejan en el sistema.

Los prestadores de servicio social son usuarios que, sin importar su perfil profesional, tienen capacidades técnicas suficientes para operar sistemas web y navegación en internet, considerando que se encuentran próximos a concluir una licenciatura y que se desenvuelven en un ambiente donde las tecnologías de la información son dominantes.

Las instituciones receptoras son aquellas del sector público y social que, de acuerdo a los lineamientos vigentes, hayan sido aprobadas por la DGOSE para registrar programas de servicio social que deriven de alguno de los ejes de acción definidos por la Universidad y que además pueden solicitar prestadores de servicio social.

La DGOSE es una dependencia universitaria que entre sus funciones tiene la responsabilidad de administrar los procesos del Servicio Social Universitario y de supervisar que estos sean ejecutados, por todos los involucrados, de acuerdo a lo establecido en los lineamientos vigentes de la Universidad.

Las Unidades Responsables de Servicio Social de facultades y escuelas están encargadas de aprobar, promover, supervisar y evaluar todos los programas de servicio social. Así como realizar el registro del inicio y término de las actividades de los prestadores de servicio social.

2.4. Restricciones (políticas regulatorias, limitaciones de hardware, seguridad, etc.)

El sistema y sus procesos deberán cumplir con una serie de restricciones de diferente naturaleza que se describen a continuación:

Deberá acotarse a lo establecido en las políticas regulatorias del Servicio Social Universitario. En la Universidad está normado por:

- a) El Reglamento General del Servicio Social de la Universidad Nacional Autónoma de México.
- b) Reglamento General de Estudios Universitarios.
- c) El Reglamento General de Exámenes.
- d) Los reglamentos de Servicio Social aprobados por los Consejos Técnicos y Académicos de Facultades, Escuelas, Institutos y Centros.

La capacidad máxima y las limitaciones del hardware del sistema se describen en la sección 2.1.6 de este documento.

Las computadoras cliente deberán contar obligatoriamente con un puerto de acceso a internet de banda ancha y recomendado al menos 2 GB de *RAM* para la ejecución del sistema.

El sistema contará con diferentes interfaces a otros sistemas y se conectará a los sistemas de Administración Escolar de DGAE, específicamente consumiendo los *Web Services* de Avance e Identidad. Los requerimientos y protocolos de conexión serán definidos por la misma DGAE a través de los manuales técnicos de los servicios web ya mencionados.

El lenguaje de programación con el que se desarrolla es PHP apoyado por el *framework* de desarrollo *Laravel* que implementa una estructura MVC, con el manejador de base de datos *MySQL*. La aplicación estará montada en un servidor *Apache HTTP*. Además se utilizará la tecnología de Firma Electrónica Avanzada (FEA).

Las restricciones de seguridad del sistema estarán dadas por aquellas que la DGTIC establece como mínimas para la instalación de cualquier sistema de cómputo en sus servidores, así mismo deberán cubrirse las recomendaciones que surjan después de las evaluaciones de seguridad ejecutadas por la misma DGTIC.

Otro aspecto a considerar, será el acceso basado en roles y el uso de contraseñas generadas por la DGOSE para los usuarios del sistema que las solicitan, así como el almacenamiento de datos a modo de bitácora que permitan contar con trazabilidad de algunas operaciones específicas que los usuarios ejecuten.

Se implementa la tecnología de Firma Electrónica Avanzada proporcionada por la DGTIC y los requerimientos se definen en los diferentes manuales de usuario publicados en su sitio web <http://www.fea.unam.mx/>

2.5. Suposiciones y dependencias (una lista de los factores que pueden afectar los requerimientos)

La efectividad del sistema puede verse afectada cuando algunas de las políticas regulatorias del servicio social sean modificadas. A su vez si las anteriores cambian, los procesos que se realizan en el servicio social pueden ser diferentes, afectando de esta manera el proceso que se haya desarrollado en el sistema.

3. Requerimientos Específicos

El SIASS Web estará conformado por los siguientes módulos, que posteriormente se describirá a detalle cada módulo:

Módulo 1: Gestión institucional (Instituciones Receptoras)

Módulo 2: Administración de programas de servicio social

Módulo 3: Gestión escolar (Prestadores de Servicio Social)

Módulo 4: Consulta de Programas

Módulo 5: Generador de reportes y extracción de información estadística

Módulo 6: Certificación electrónica del término de servicio social.

3.1. Módulo 1: Gestión Institucional (Instituciones Receptoras)

En este módulo se realizarán las acciones de registro, aprobación, actualización y baja de los usuarios pertenecientes a las instituciones receptoras. Los usuarios de IR se registrarán por su cuenta y el personal del Departamento de Registro y

Seguimiento de la DGOSE aprobará la emisión de contraseñas si se cumple con los lineamientos vigentes.

Los usuarios de las IR deberán ingresar la siguiente información para el registro:

- a) El sector al que pertenece la Institución (público o social).
- b) Nombre de la Institución.
- c) Nombre de la Dependencia.
- d) Rol de servicio social (Coordinador o Responsable de programa).
- e) Domicilio laboral.
- f) Teléfono de oficina.
- g) Nombre y correo del usuario que se registra.
- h) Oficio de firmas (proporcionado por la DGOSE) en formato PDF y máx 2 Mb.
- i) Formato de programa y acta constitutiva (únicamente IR del sector social) en formato PDF y máx. 2 Mb.

Para usuarios de tipo DGOSE, el sistema mostrará la información que los usuarios han registrado, incluidos los archivos PDF; esto con la finalidad de dictaminar si se genera o no una contraseña de acceso para el usuario en cuestión.

El sistema enviará diferentes correos electrónicos para el registro generando un folio de seguimiento, para la emisión de contraseña cuando el usuario sea aprobado; así como correos cuando exista un dictamen de “Rechazado” o asignación de clave a los programas registrados.

El registro de instituciones receptoras de servicio social, se describe mediante el siguiente diagrama de flujo:

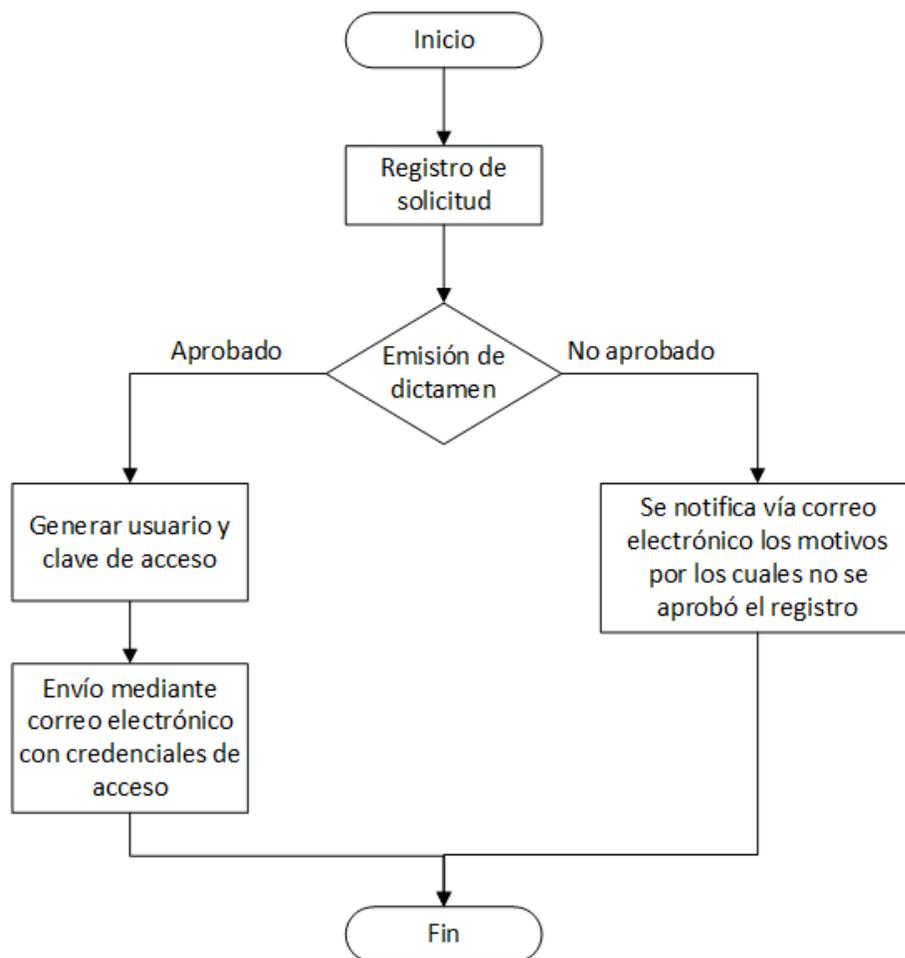


Ilustración 34 Flujo de operación. Fuente: Elaboración propia

3.2. Módulo 2: Administración de programas de servicio social

En este módulo se podrán realizar los procesos de registro, validación y publicación de programas de servicio social.

Una vez que los usuarios IR han adquirido su contraseña de acceso podrán registrar por cuenta propia los programas de servicio social que deseen. Los programas deben cumplir con los lineamientos establecidos en el Reglamento General de Servicio social o de lo contrario no serán validados ni publicados.

La información de los proyectos de servicio social estará estructurada por los siguientes bloques de información:

- a) Institución y Dependencia ejecutora
 - Nombre de la Institución y dependencia ejecutora.

- Nombre, teléfono y correo electrónico del coordinador y del responsable del proyecto de servicio social.
 - Domicilio laboral.
- b) Información del proyecto de servicio social
- Eje de Acción.
 - Nombre del programa.
 - Objetivo del programa.
- c) Ubicación geográfica del programa de servicio social
- Entidad federativa
 - Delegación o municipio
 - Colonia o localidad
 - Código postal
 - Calle y número
- d) Información de la población beneficiada, el tipo de atención y los grupos vulnerables a atender.
- e) Apoyos para el prestador, turno y horarios
- f) Información del perfil profesional de los prestadores
- Carreras solicitadas.
 - Número de prestadores solicitados por carrera.
 - Registro de al menos 5 actividades por cada carrera.

Después de que el programa es registrado por el usuario de la IR, los usuarios URSS determinarán que proyectos de servicio social son válidos para su escuela o facultad y carreras que tienen a su cargo. Recordando que la IR registra el programa y elige que carreras solicita, de acuerdo con cada carrera solicitada le llegará a cada usuario URSS la notificación para evaluar ese programa para las carreras que le compete administrar en materia de servicio social.

3.3. Módulo 3: Gestión Escolar (Prestadores de Servicio Social)

En este módulo se podrá realizar el registro, cambios y bajas de los prestadores de servicio social así como la liberación de aquellos que cumplan los requisitos establecidos en los lineamientos generales.

Se tendrán cuatro modalidades de prestación de servicio social por:

- a) Seis meses o “convencional”.

- b) Opción a titulación.
- c) Artículo 52, que se refiere al de la "Ley Reglamentaria de los Artículos 4º y 5º Constitucionales, relativo al Ejercicio de las Profesiones en el Distrito Federal y Territorios Federales" que establece que "Todos los estudiantes de las profesiones a que se refiere esta ley, así como los profesionistas no mayores de 60 años, no impedidos por enfermedad grave, ejerzan o no, deberán prestar el servicio social en los términos de esta ley." (Facultad de Ingeniería, s.f.).
- d) Artículo 91, que se refiere al de la "Ley Reglamentaria de los Artículos 4º y 5º Constitucionales, relativo al Ejercicio de las Profesiones en el Distrito Federal y Territorios Federales" que establece que "los estudiantes y profesionistas trabajadores de la Federación y del Gobierno del Distrito Federal no estarán obligados a prestar ningún servicio social distinto del desempeño de sus funciones. El que presten voluntariamente dará lugar a que se haga la anotación respectiva en su hoja de servicios."

La información del prestador de servicio que se registra, se divide en diferentes categorías únicamente para mantener un orden y una estructura:

- a) Información escolar
 - Número de cuenta.
 - Clave y nombre de la facultad o escuela.
 - Clave y nombre de la carrera.
 - Porcentaje de créditos.
 - Promedio.
 - Modalidad de estudios (escolarizado, universidad abierta o educación a distancia).
- b) Información personal
 - Apellido paterno.
 - Apellido materno.
 - CURP.
 - Género.
- c) Información de ubicación y contacto
 - Clave y nombre de la entidad federativa.
 - Clave y nombre de la delegación o municipio.
 - Código postal.
 - Colonia o localidad.
 - Calle y número.
 - Teléfono de casa.
 - Teléfono celular.
 - Teléfono de oficina.
 - Correo electrónico.
 - Correo electrónico alterno.

- d) Información del programa y la modalidad de servicio social
- Modalidad de servicio social.
 - Clave y nombre del programa de servicio social (si es el caso).
 - Clave y nombre de la Institución Receptora.
 - Clave y nombre de la Dependencia ejecutora.
 - Fecha de inicio.
 - Fecha de término.
 - Fecha de certificación.

Dentro de este módulo se implementará la comunicación a través de los *web services* de la DGAE y que son mencionados en el punto 2.4 de este documento. Los datos que se obtendrán serán almacenados de manera local en el propio SIASS para su uso posterior.

Los datos enviados serán:

- a) Número de cuenta
- b) Clave de la facultad o escuela
- c) Clave de la carrera
- d) Nivel académico

Por cada prestador de servicio social se generara un identificador único el cual estará conformado por el número de cuenta concatenado a la clave de la facultad o escuela y la carrera. Esto permitirá que los alumnos que cursan más de una carrera puedan ser registrados en el sistema utilizando el mismo número de cuenta.

Ejemplo:

Número de cuenta	Clave de la Facultad/Escuela	Clave de la carrera
304047181	004	303

Las claves de las facultades, escuelas y carreras serán definidas por la DGAE que serán proporcionadas en forma de catálogos en archivos de texto y el sistema deberá ser capaz de leerlos e interpretarlos para mantener actualizada la base de datos.

Las URSS son las únicas con la facultad de registrar prestadores de servicio social. Los usuarios de tipo DGOSE serán capaces de ejecutar esta acción, aunque en la práctica solo se realice en contadas excepciones.

Para la conclusión del servicio social, se deberán generar certificados electrónicos utilizando la tecnología de Firma Electrónica Avanzada y proporcionará un servicio web "Término de servicio social", el cual será consultado por la DGAE para conocer la situación actual respecto al servicio social de un estudiante o egresado (módulo 6).

Cuando el proceso de servicio social no sea el ideal, es decir que no se termine en el periodo indicado y dentro del programa registrado inicialmente, el sistema

emitirá distintas notificaciones dentro del sitio web en forma de ventanas emergentes y en el listado de prestadores registrados, las notificaciones pueden ser las siguientes:

- a) Notificación de término del periodo de servicio social registrado.
- b) Notificación 1 año después de la fecha de inicio registrada.
- c) Notificación 1 año y 6 meses después de la fecha de inicio registrada.
- d) Notificación 2 años después de la fecha de inicio registrada. Esta última notificación también dará dos opciones únicas para el estatus del prestador:
 - Concluido
 - Baja del servicio social. El alumno deberá ser registrado en otro programa vigente o en una modalidad distinta para poder liberar y certificar su servicio social.

3.4. Módulo 4: Consulta de Programas

En este módulo el sistema permitirá realizar la consulta de programas de servicio social registrados en el sistema.

Con el acceso basado en roles se determinará la forma en la que se muestran los programas, los usuarios de tipo URSS podrán ver todos los programas registrados que soliciten las carreras que tiene a su cargo, los usuarios IR únicamente podrán ver los programas registrados a nombre de su Dependencia Ejecutora y los usuarios DGOSE deberán tener visibles la totalidad de los programas registrados.

Prestadores de Servicio Social

Los alumnos o futuros prestadores de servicio social contarán con un sub módulo aparte que estará conformado por un formulario donde se ingresará el número de cuenta, la escuela o facultad y la carrera deseada. Este formulario devolverá un listado con todos los programas de servicio social aprobados que correspondan con la información ingresada previamente. Por ejemplo un alumno que cursa la carrera de Arquitectura en la Facultad de Arquitectura podrá visualizar los programas aprobados por el Responsable de Servicio Social de la Facultad de Arquitectura y no así los programas aprobados por el responsable de la Facultad de Estudios Superiores Aragón o Acatlán, planteles donde también se imparte dicha carrera.

Una vez mostrados los programas disponibles, el alumno podrá filtrar esta búsqueda de acuerdo a diversos parámetros como lo son:

- a) Eje de acción.
- b) Sector de la Institución receptora (público y social).
- c) Ubicación geográfica del programa.

- d) Apoyos al prestador.
- e) Turno y días de asistencia.
- f) Tipos de apoyo al prestador.

Unidades Responsables de Servicio Social

Los responsables de servicio social de las facultades y escuelas podrán consultar la información de los proyectos de servicio social donde se hayan solicitado las carreras que tienen bajo su administración.

Así mismo sólo podrán tener acceso a la información de los prestadores que pertenezcan a su facultad o escuela y a las carreras que tienen bajo su administración.

Usuario Institución Receptora

Este usuario podrá realizar búsquedas únicamente de los programas de servicio social que hayan sido registrados para la Dependencia Ejecutora a la que pertenece

Dentro de la información propia de cada programa se podrán consultar el estatus de la validación que emiten las URSS, es decir si el proyecto se encuentra en revisión, aprobado o rechazado. En caso de estar rechazado también se mostrarán los motivos del rechazo del programa y los datos de contacto del evaluador.

3.5. Módulo 5: Generador de reportes y extracción de información estadística

Diseñado para proporcionar una interfaz gráfica que permita, a las URSS y al personal administrativo de la DGOSE, la extracción de reportes gráficos e información estadística de los datos almacenados más relevantes de acuerdo a lo definido por la DGOSE.

En un inicio las URSS podrán obtener gráficas e información sintetizada para:

- a) Programas registrados.
- b) Demanda de prestadores.
- c) Registro de prestadores.
- d) Prestadores liberados.

Adicionalmente los usuarios DGOSE podrán consultar las siguientes estadísticas:

- a) Programas Anuales, divididos por sector (público y social)
- b) Programas propios de alguna dependencia de la UNAM.

Las estadísticas para los programas registrados serán agrupadas de acuerdo al sector o al eje de acción del programa y filtradas con los elementos de la siguiente lista:

- a) Año y mes de registro.

- b) Entidad federativa y delegación o municipio.
- c) Si el programa se ejecuta dentro de algún campus UNAM.
- d) Eje temático.
- e) Población beneficiada.
- f) Tipos de apoyo al prestador.
- g) Sector.
- h) Institución Receptora.
- i) Dependencia Ejecutora.

Para la sección de Demanda de prestadores se agruparán los resultados por carreras o ejes de acción y se podrán filtrar de la siguiente manera:

- a) Año y mes de registro.
- b) Entidad federativa y delegación o municipio.
- c) Eje temático.
- d) Población beneficiada.
- e) Carrera.
- f) Sector.
- g) Institución Receptora.
- h) Dependencia Ejecutora.

Para los prestadores registrados se agruparán por escuela/facultad o por eje de acción, y se podrá utilizar filtros de acuerdo a los siguientes criterios:

- a) Año y mes de registro.
- b) Modalidad de estudios (escolarizado, abierto, a distancia).
- c) Escuela o facultad.
- d) Carrera.
- e) Modalidad de servicio social (convencional, artículo 91, artículo 52 y opción de titulación).

La sección de estadísticas para prestadores liberados mostrará el total dividido por género (hombres y mujeres) y con la posibilidad de filtrar de la siguiente forma:

- a) Año y mes de registro.
- b) Modalidad de estudios.
- c) Escuela o facultad.
- d) Carrera.
- e) Modalidad de estudios (escolarizado, abierto, a distancia).
- f) Estatus del servicio social (Liberado/certificado, concluido, baja o en proceso).

3.6. Módulo 6: Certificación electrónica del término de servicio social

Como ya se había mencionado anteriormente, la emisión de certificados digitales de término de servicio social se realizará implementando la tecnología de Firma Electrónica Avanzada.

La Dirección de Servicio Social, a través del titular de la Dirección General de Orientación y Servicios Educativos se encargará de certificar electrónicamente a los prestadores de servicio social que se encuentren con un estatus de “Concluido” en el sistema; este estatus será asignado por las URSS a través del sistema una vez que consideren que el prestador ha cumplido con todas sus actividades y trámites.

Para poder generar el certificado electrónico, el sistema contará con un apartado exclusivo para el Director General, en turno, de la DGOSE en donde únicamente se visualizarán los prestadores con el estatus de “Concluido” para ejecutar la acción de firmado electrónico a dichos prestadores.

La acción del firmado electrónico será ejecutada en esta misma sección del sistema, que contará con el módulo embebido correspondiente. Las definiciones y características mínimas para ejecutar dicho módulo serán proporcionadas por la DGTIC a través del Departamento de Firma Electrónica Avanzada de la UNAM.

Cuando se ejecute la acción, el módulo embebido solicitará la FEA del titular de la DGOSE para entonces generar el certificado de cada uno de los prestadores mostrados en la lista.

Una vez que se haya certificado el término del servicio social ya no será posible realizar modificaciones, dar de baja o crear un nuevo registro de la misma carrera para el prestador. El registro certificado será definitivo para fines de titulación de la carrera que se indica.

El identificador de cada certificado electrónico de término de servicio social será generado a partir del número de cuenta, la clave de la facultad o escuela y la carrera del prestador de servicio social.

De acuerdo a los lineamientos establecidos por el Departamento de la FES, la DGOSE tiene la responsabilidad de resguardar y respaldar cada uno de los certificados electrónicos emitidos, por lo que el diseño del sistema debe contemplar este requerimiento.

Finalmente, el sistema deberá contar con un servicio web de consulta, donde inicialmente la DGAE podrá obtener información en tiempo real del estatus de servicio social de los prestadores. Cuando un prestador ya cuenta con certificado electrónico de liberación entonces la DGAE podrá dar por completado el requisito de servicio social para la titulación.

Se planea a mediano plazo generar un servicio web similar para las URSS que además contará con un histórico para los casos donde el prestador cause baja y tenga más de un programa o modalidad de servicio social en sus registros.

Descripción del Diseño del Software (*Software Design Description*)

1. Introducción

1.1. Propósito

El propósito del presente documento es proponer a la Dirección General de Orientación y Servicios Educativos (DGOSE) el diseño del Sistema de Información Automatizada de Servicio Social (SIASS), basado en la especificación de los requerimientos de software (SRS por sus siglas en inglés). Al finalizar el documento se mostrarán la arquitectura, los módulos, funcionalidades y componentes que se presentan en éste diseño.

Para realizar el presente documento se siguieron las prácticas recomendadas por la IEEE en el estándar 1016 Descripción del Diseño de Software (IEEE Computer Society, 2009). Se tomaron algunos de los *viewpoints* que contiene el estándar los cuales nos ayudan a describir los puntos que queremos tratar.

1.2 Alcance

Este documento va dirigido a los desarrolladores de sistemas, quienes habrán de programar e implementar el sistema, a las personas encargadas de calidad puesto que habrán de probar el software y a aquellos que en un futuro den mantenimiento al sistema.

1.3 Authorship

La información de este documento es elaborada por los estudiantes que presentan la tesis “Desarrollo de un Sistema Integral Para la Administración de Servicio Social”. La información presentada en este documento es una propuesta de diseño, por lo que se puede hacer pública y sin infringir la confidencialidad o el uso de derechos de autor.

1.4 Referencias

Este documento está elaborado basándose en el estándar 1016-2009 publicado por el Instituto de Ingenieros en Electricidad y Electrónica (IEEE por sus siglas en inglés) titulado Descripción del Diseño de Software (*Software Design Description SDD*) (IEEE Computer Society, 2009).

IEEE Computer Society. (2009). Std. 1016-2009. Recuperado el Junio de 2011, de IEEE Xplore.

1.5 Contexto

La Universidad Nacional Autónoma de México cuenta con un sistema para la administración del servicio social, sin embargo con la finalidad de incentivar el uso

de las tecnologías de la información dentro la Universidad, y ante la necesidad de actualizar los procesos, se presenta la oportunidad de desarrollar un nuevo sistema para el servicio social. Con la implementación de estas nuevas acciones y el incremento del uso de las TIC, se busca que el SIASS implemente lenguajes de programación, *frameworks* de desarrollo y componentes en sus versiones más recientes de tal forma que esto permita tener un software funcional, amigable para el usuario y bien documentado.

1.6 Lenguajes y tecnologías

SDD: UML
Desarrollo:
• Lenguaje de Programación PHP v5.6
• Framework de Desarrollo Laravel v4
• Manejador de base de datos MySql v5
• Servidor HTTP Apache v2.4
• Componente de Tecnología de Firma Electrónica Avanzada (FEA).

Tabla 5 Lenguajes y tecnologías del SIASS. Fuente: elaboración propia.

2. Identificación de los interesados y áreas de interés

Los interesados aquí presentados son las personas interesadas en el diseño, sin embargo, tiene un símil respecto al desarrollo ágil de Scrum, por lo cual se menciona el interesado de diseño, con el interesado de Scrum que le corresponde.

- Los usuarios administrativos del Servicio Social en la DGOSE ya que ellos son quienes solicitan el sistema, y actúan como el “Cliente” o en desarrollo Scrum como *Product owner*.
- El equipo de desarrolladores del sistema, que son los del *Scrum team*.
- Los administradores del proyecto, que son los *Scrum master*.
- La DGAE a través de la Subdirección de Sistemas de Registro Escolar que provee y consume *web services* con el sistema SIASS web.

De acuerdo a la tabla 1 *Summary of design viewpoints* del Std.1016 (IEEE Computer Society, 2009, p. 13), se eligieron las áreas de interés más significativos para este documento, con el o los interesados correspondientes.

Áreas de interés	Interesados	Descripción
Servicios del sistema y usuarios	DGOSE	La DGOSE encabeza este nuevo sistema, por lo tanto está interesado en la construcción y en los servicios que ofrecerá el SIASS a los diversos usuarios.
	Desarrolladores	Debe conocer los servicios que debe dar el SIASS para poder desarrollarlos.

	Administradores del proyecto	Diseñarán la planeación del ciclo de vida de desarrollo del software, lo que implica conocer aquello que la DGOSE requiere para el sistema, y mediar con los desarrolladores a fin de cumplir con lo establecido en la planeación.
Arquitectura	DGOSE, Administradores del proyecto.	Se sugiere y se establece como requerimiento implementar la tecnología de Modelo-Vista-Controlador (MVC)
	Desarrolladores	Implementar el desarrollo del sistema utilizando la arquitectura MVC.
Componentes	Administradores del sistema	Ya que se hará uso de <i>web services</i> con DGAE, los administradores del sistema deberán ser los facilitadores de la información y requerimientos hacia los desarrolladores para llevar a cabo esta implementación.
	Desarrolladores	Desarrollaran un <i>web service</i> proveedor para exponer un servicio de SIASS, así como un cliente para consumir un servicio de DGAE.
Interconexión	Administradores del sistema	Teniendo en cuenta que existirá una comunicación de los <i>web services</i> de DGAE y de SIASS, es tarea del Administrador conocer la forma de comunicación, para poder gestionarla.
	Desarrolladores	Deben conocer la estructura y las reglas de la comunicación de los <i>web services</i> de DGAE y de SIASS, ya sea para exponer o consumir los servicios.
Información de persistencia	Administradores del sistema	Los administradores deben contemplar quién y cómo se administrará la base de datos.
	Desarrolladores	Los desarrolladores, harán uso de la base de datos, para hacer inserciones, consultas u otras operaciones.
Acceso al servicio	Administradores del sistema	Porque se define cómo será la interfaz de acceso al sistema, de los usuarios.
	Desarrolladores	Programarán las interfaces que permitirán el funcionamiento del sistema.
	DGOSE	Son los usuarios, los que accederán al sistema.

Tabla 6 Resumen de design concerns correspondientes al SIASS. Adaptado de (IEEE Computer Society, 2009, p. 13).

3. Puntos de vista

3.1. Punto de vista de Contexto

Éste punto de vista considera como áreas de interés a los servicios del sistema y usuarios. Se describen gráficamente las funcionalidades que ofrecerá el SIASS.

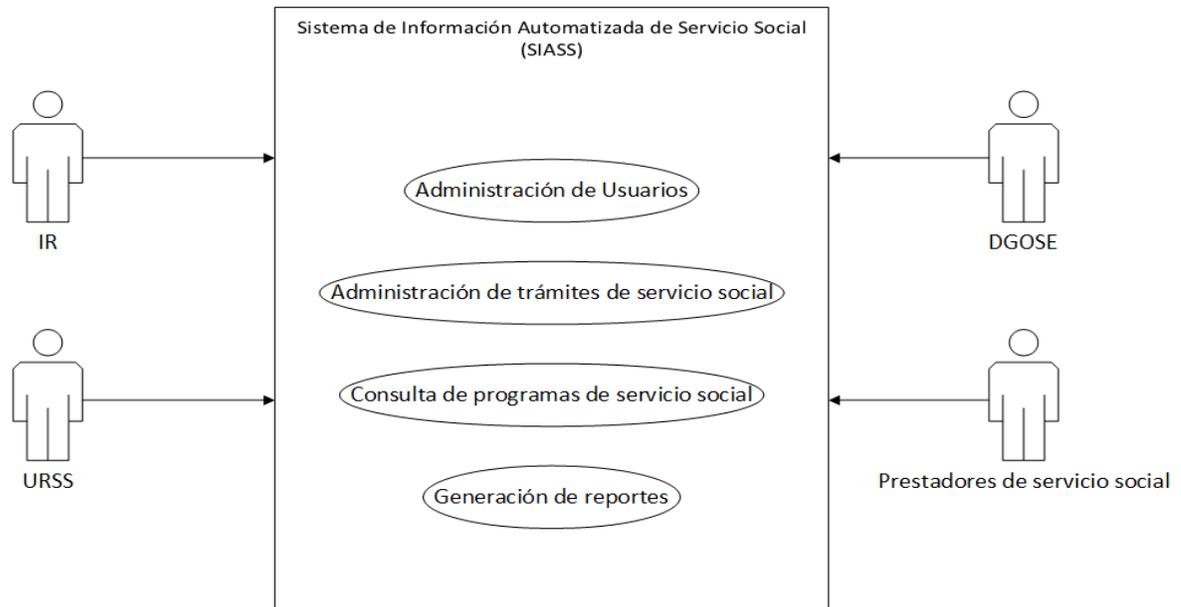


Ilustración 35 Funcionalidades del SIASS. Fuente: elaboración propia.

El desarrollo del sistema está contemplado en 6 módulos diferentes

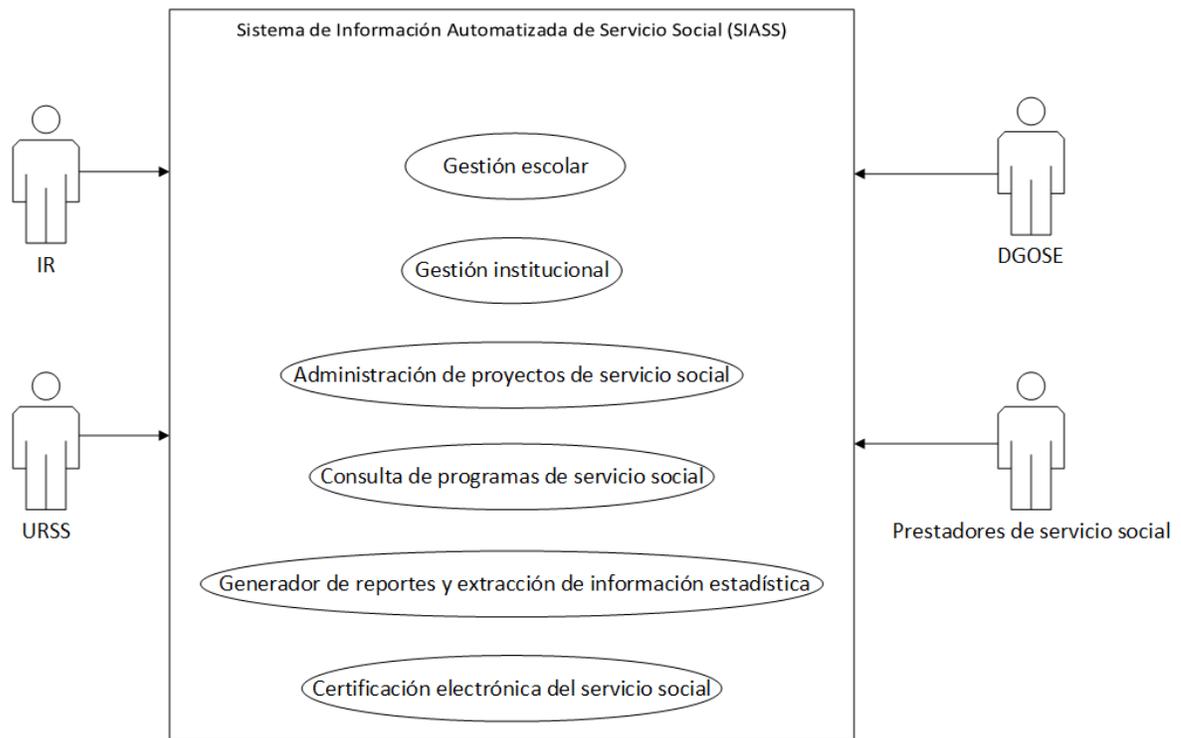


Ilustración 36 Módulos del SIASS. Fuente: elaboración propia.

Por cada módulo se presenta el diagrama de caso de uso, para una comprensión más amplia.

Módulo 1. Gestión Institucional

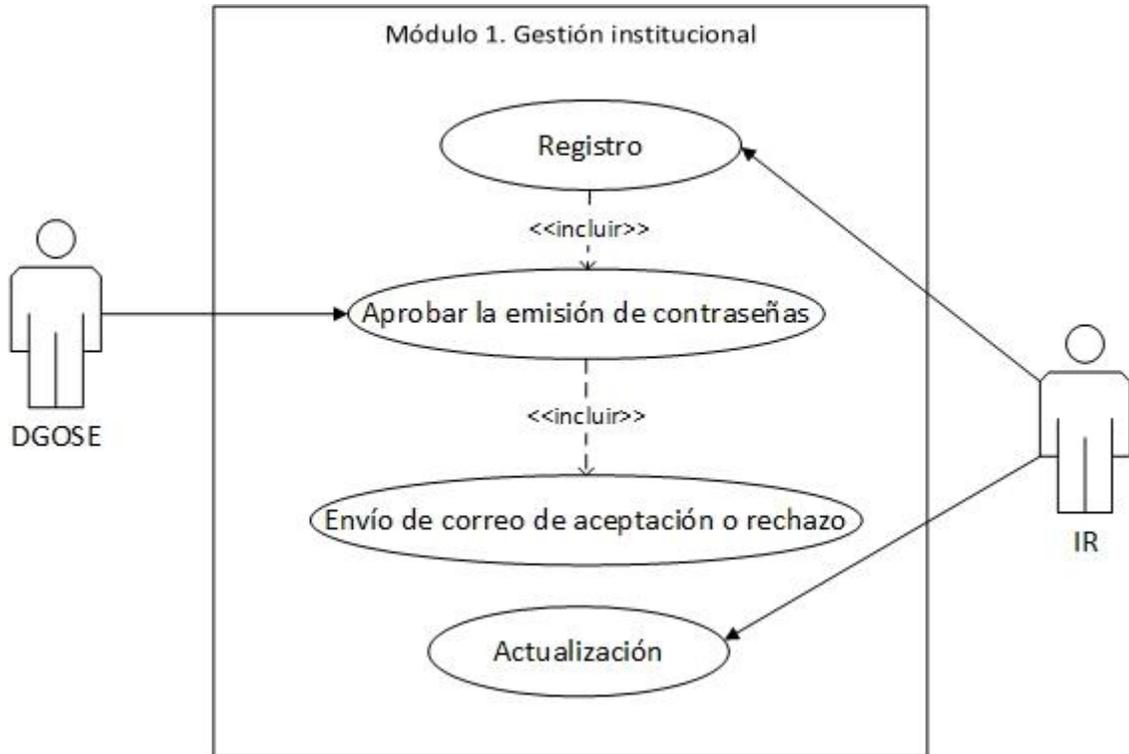


Ilustración 37 Gestión institucional. Fuente: elaboración propia.

Módulo 2. Administración de proyectos de servicio social

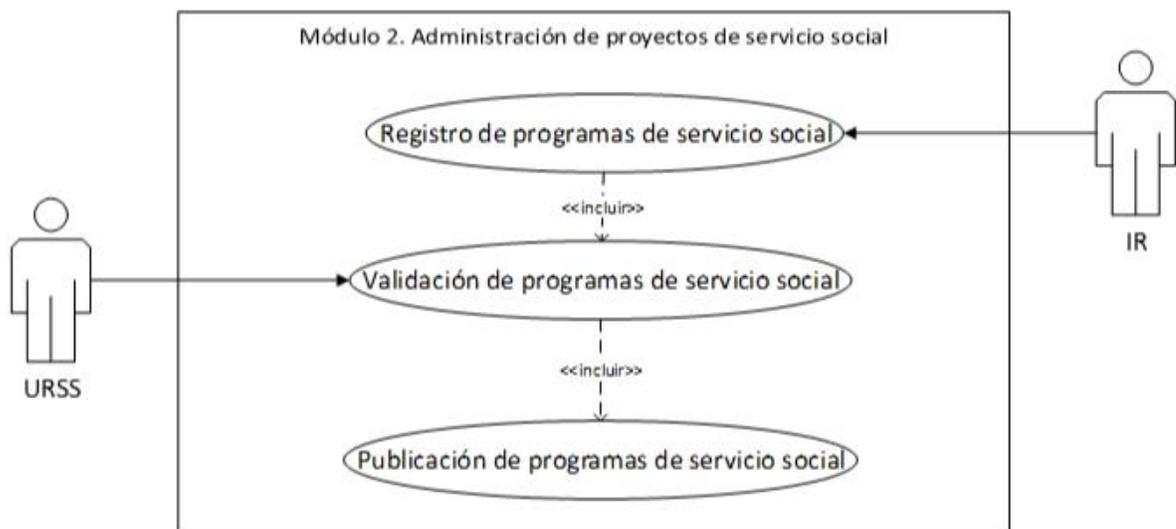


Ilustración 38 Administración de programas de servicio social. Fuente: elaboración propia.

Módulo 3. Gestión Escolar

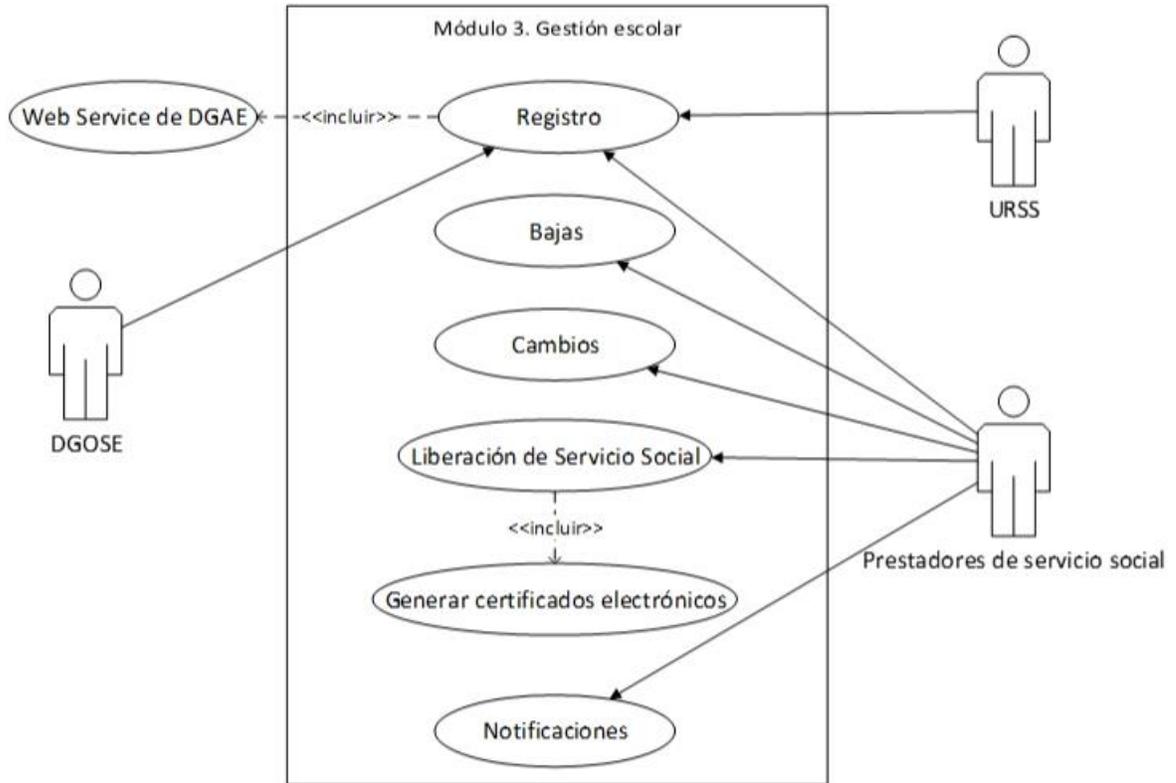


Ilustración 39 Gestión escolar. Fuente: elaboración propia.

Módulo 4. Consulta de programas de servicio social

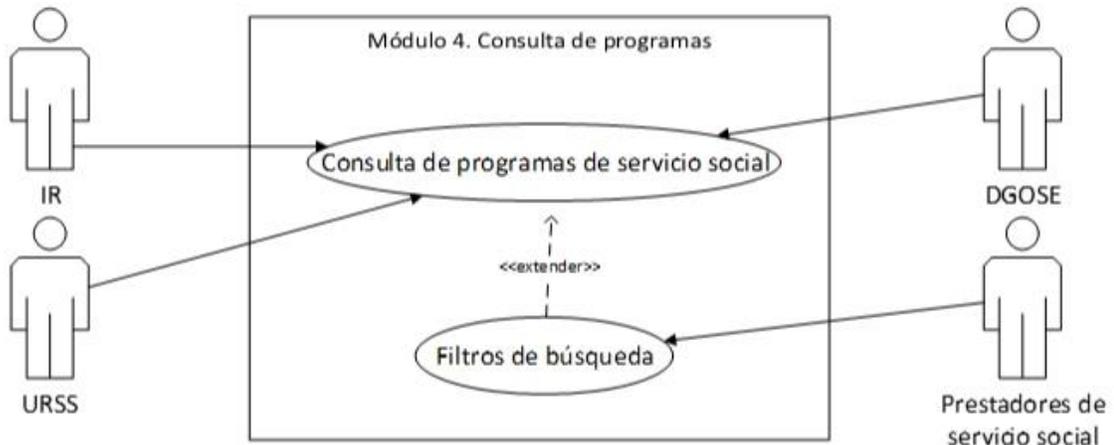


Ilustración 40 Consulta de programas. Fuente: elaboración propia.

Módulo 5. Generador de reportes y extracción de información estadística.

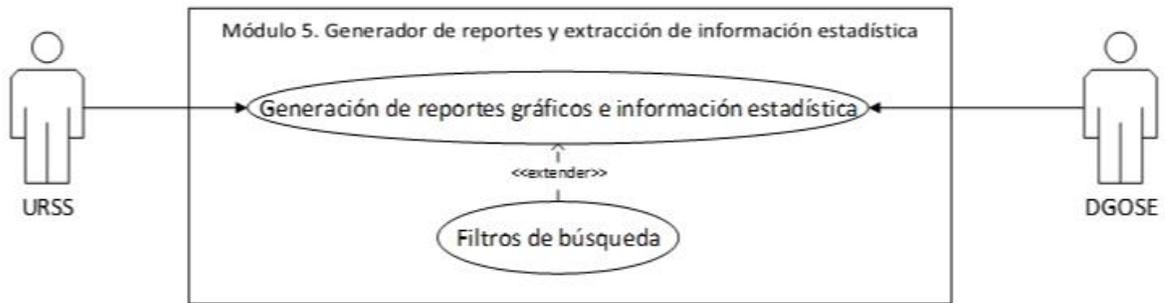


Ilustración 41 Generador de reportes. Fuente: elaboración propia.

Módulo 6. Certificación electrónica del término de servicio social

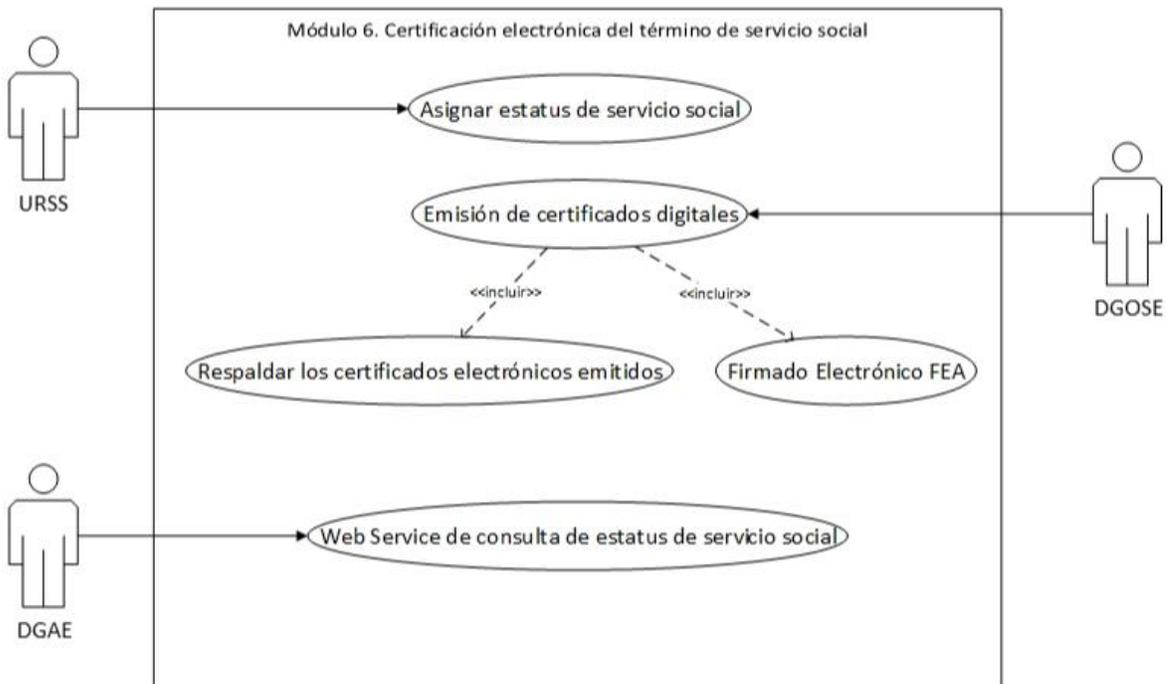


Ilustración 42 Certificación electrónica. Fuente: elaboración propia.

Adicionalmente se incluye un diagrama de flujo que muestra en forma resumida el flujo del sistema.

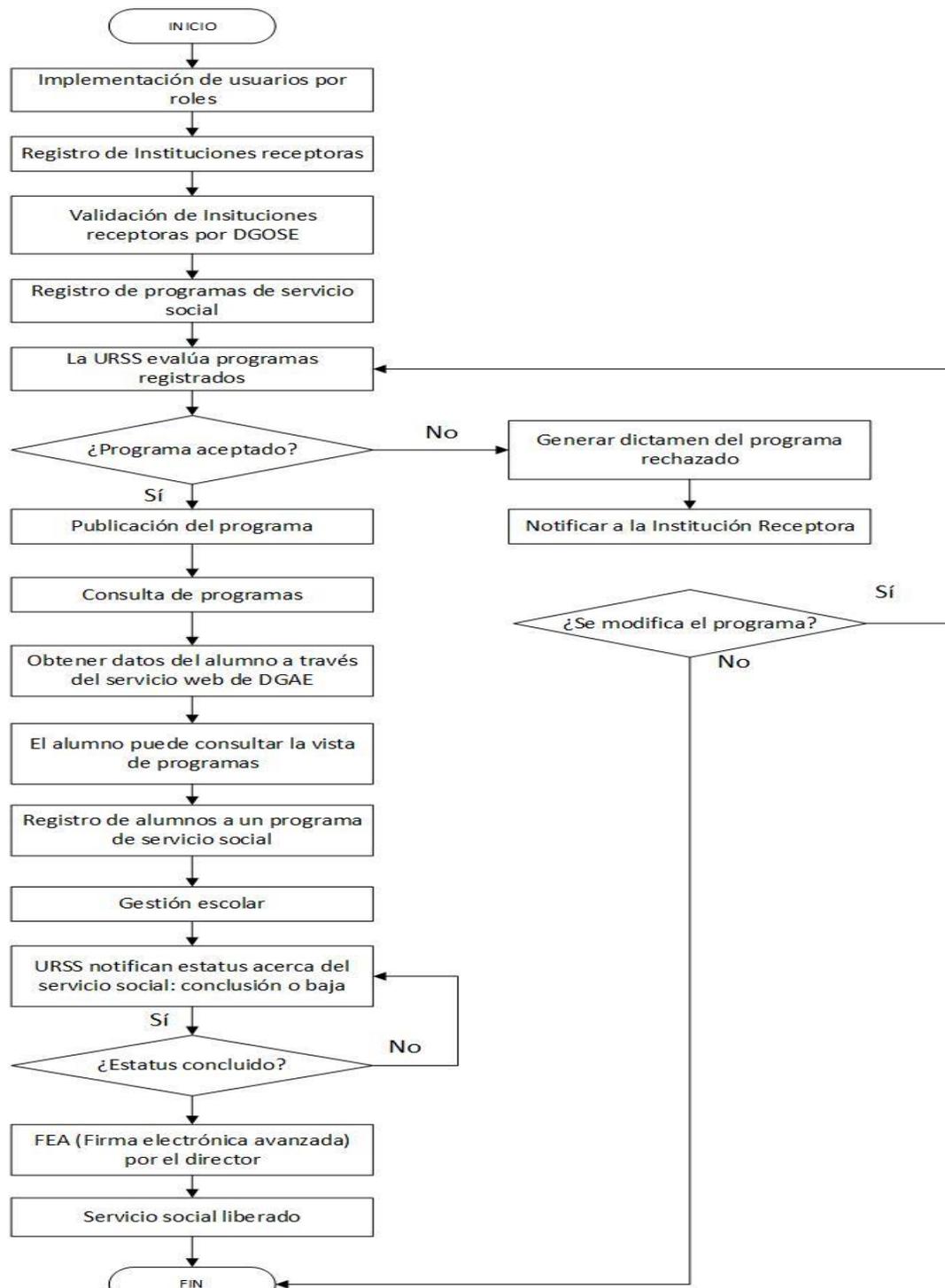


Ilustración 43 Diagrama de flujo del SIASS. Fuente: elaboración propia

3.2. Punto de vista de Composición

Se describe la estructura y los componentes del sistema. Se considera el área de interés de la arquitectura que permitirá a los desarrolladores y a aquellos encargados del mantenimiento del sistema, localizar las funcionalidades, y tener una mejor perspectiva de la arquitectura del sistema.

La arquitectura Cliente – Servidor:

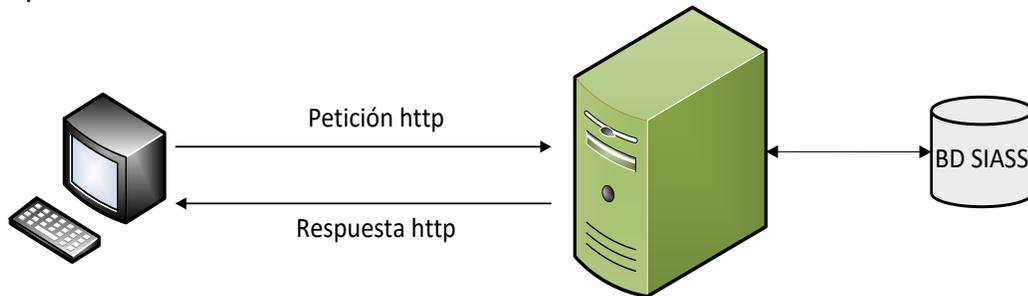


Ilustración 44 Arquitectura cliente servidor del SIASS. Fuente: elaboración propia.

La arquitectura Modelo-Vista-Controlador:

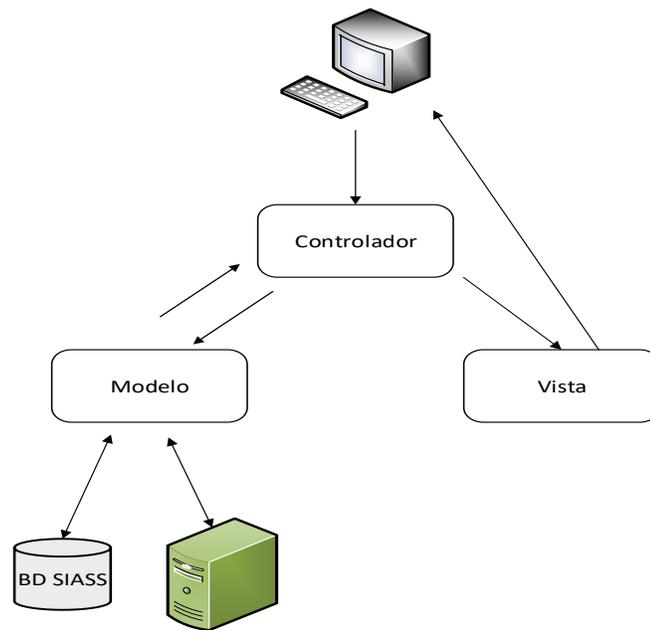


Ilustración 45 MVC del SIASS. Fuente: elaboración propia.

3.3 Punto de vista de Dependencia

Se considera éste punto de vista para el área de interés de Interconexión. El diagrama de componentes para los *web services* que estarán interactuando es el siguiente:

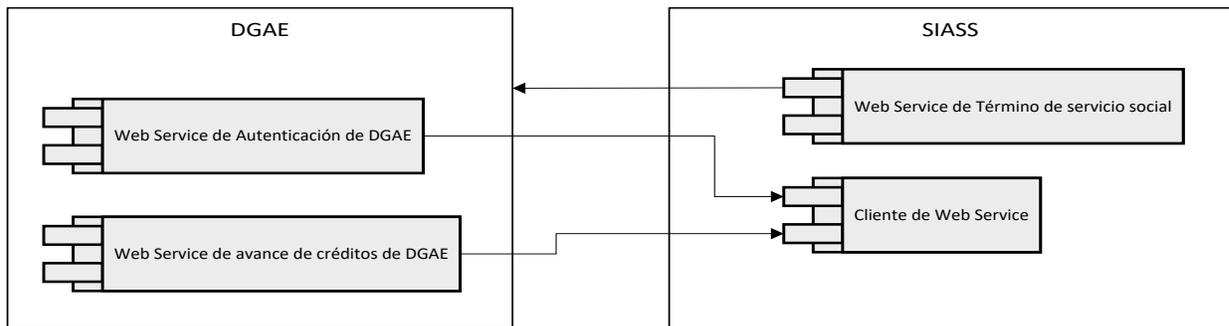


Ilustración 46 Diagrama de componentes del SIASS con DGAE. Fuente: elaboración propia

Otro aspecto en la dependencia es la relación entre los módulos y la información que comparten.

3.4 Punto de vista de Información con distribución de datos y su volumen

Éste punto de vista comprende el diagrama entidad relación de la base de datos, en el que se presentan algunas tablas significativas del sistema, ya que por causas de confidencialidad no se debe mostrar toda la base de datos. Sin embargo, el presente diagrama muestra un panorama claro de lo que se representa.

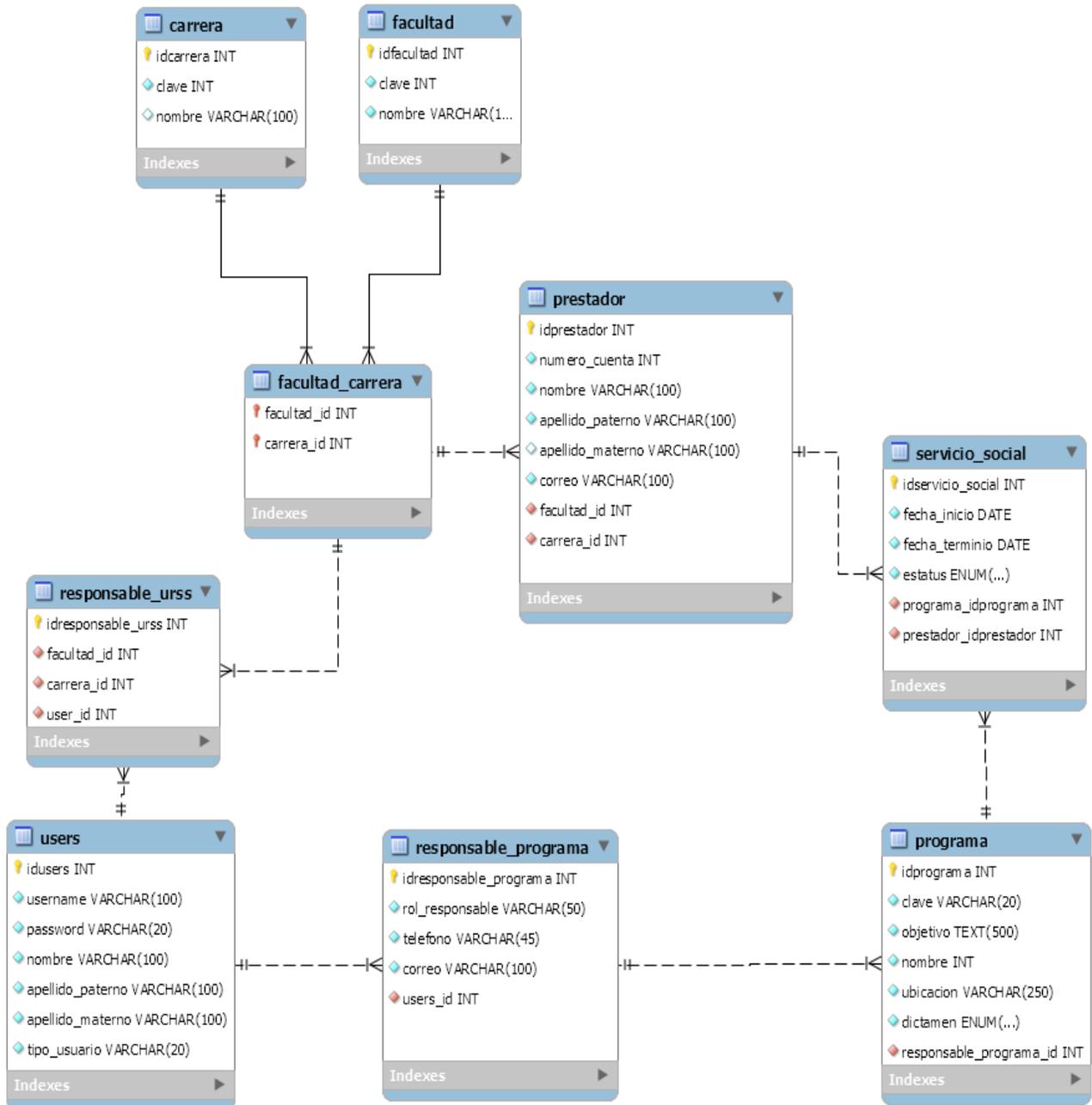


Ilustración 47 Diagrama entidad relación del SIASS. Fuente: elaboración propia

3.5 Punto de vista de Interfaz

Con el punto de vista de la interfaz se cubre el área de interés de acceso al servicio, que nos brinda información acerca de cómo acceder a los servicios del sistema. Se presentan prototipos de las vistas para la organización de las pantallas (Ilustraciones 48 a 52).

Para poder realizar la consulta, el alumno debe ingresar sus datos tales como son número de cuenta, escuela o facultad y la carrera que cursa, con esto el SIASS hará un filtro para mostrar únicamente los programas aprobados que corresponden a la carrera del alumno.



Ilustración 48 Consulta de programas (1). Fuente: elaboración propia.

Cuando se recupere la información se mostrará una pantalla con los programas encontrados, posteriormente el alumno podrá ingresar información adicional para acotar aún más la búsqueda. Los criterios de búsqueda serán programas dentro de la UNAM, de instituciones públicas o del sector social.



Ilustración 49 Consulta de programas (2). Fuente: elaboración propia.

Posteriormente el alumno podrá consultar el detalle de cada programa mostrado como se muestra en las siguientes vistas:

Sistema de Información Automatizada de Servicio Social

Dependencia ejecutora

Nombre Servicio de Administración Tributaria, SHCP

Área Administración de servicios del D.F.

Responsable del proyecto: Ing. Miguel Ángel León

Teléfono y correo 1023 – 1536, Roquel.diaz@sat.gob.mx

Domicilio: Av. Cuauhtémoc, No. 602, 1er. Piso, Delegación Benito Juárez, Col. Narvarte; C.P. 03020, D.F.

Información del proyecto de Servicio Social

Campo de acción: Modernización y automatización de la gestión administrativa

Nombre Apoyo técnico y administrativo al servicio de administración tributaria

Objetivo Coordinar en la actualización de la normatividad para la recaudación de las contribuciones federales a través de la correcta aplicación de la legislación.

Información de prestadores de servicio social

Carrera Contaduría

Número de prestadores: 3

Actividades Análisis Demográfico

Carrera	Prestadores registrados	Actividades
Informática	2	Análisis y diseño de datos - análisis de requerimientos
Administración	1	Diseño de mapas conceptuales - diseño organizacional

¿Las actividades se realizan dentro de las instalaciones de la Sí No

Ilustración 50 Consulta de programas (3). Fuente: elaboración propia.

Ubicación geográfica del proyecto

Entidad: México D.F.

Delegación/Municipio: Xochimilco

Colonia/Localidad: Huichapan

C.P.: 16030

Calle y número: 14 de julio #67

Apoyos y horarios

Día	Requerido	Día	Requerido
Lunes	<input checked="" type="checkbox"/>	Martes	<input checked="" type="checkbox"/>
Miércoles	<input checked="" type="checkbox"/>	Jueves	<input checked="" type="checkbox"/>
Viernes	<input checked="" type="checkbox"/>	Fines de semana	<input type="checkbox"/>

Horario: Matutino Vespertino Mixto

Apoyos: Académicos Económicos Otros

Monto: \$2000

Patrocinador: Servicio de Administración Tributaria, SHCP

Responsable de Servicio Social

Nombre: Lic. Diego Díaz Reyes

Cargo: Director de Recursos Humanos

Profesión: Licenciado en Administración

Teléfono: 56 84 75 12

Correo Electrónico: correo@sat.df.com

Domicilio: 14 de julio #67 Colonia. Huichapan Del. Xochimilco

Ilustración 51 Consulta de programas (4). Fuente: elaboración propia.

Adicionalmente para el módulo de Gestión Escolar, los usuarios de tipo URSS podrán consultar la información escolar de los alumnos a través del *web service* proporcionado por la DGAE proporcionando el número de cuenta, la escuela o facultad y la carrera, la vista resultante con la información quedaría de la siguiente manera:

Sistema de Información Automatizada de Servicio Social

Información personal	
Nombre (s):	Sergio
Apellido paterno:	Martínez
Apellido materno:	Cuevas
Género:	Masculino

Información académica	
Número de cuenta:	097111197
Facultad / escuela:	110 - Facultad de Ingeniería
Carrera:	011 - Ingeniería en Computación
Porcentaje de créditos:	79.41 %
Promedio:	8.47
Modalidad de estudios:	Sistema escolarizado

Información del proyecto de servicio social	
Programa estratégico:	Salud
Dependencia ejecutora:	Servicios de Administración Tributaria, SHCP
Área:	Administración de Servicios del D.F.

Información respecto al servicio social	
Modalidad de registro:	Servicio social 6 meses
Clave del proyecto:	2013-012-/1-1012
Nombre del proyecto:	Apoyo técnico y administrativo al servicio de administración
Fecha de inicio:	04-10-2013
Fecha de termino:	04-04-2013
Servicio social:	<input checked="" type="radio"/> Certificado Fecha de certificación: 25-04-2013
Estado:	Concluido

Ilustración 52 Registro de prestadores. Fuente: elaboración propia.

Finalmente, las vistas para el registro de usuarios tendrían diferentes características de acuerdo al tipo de usuario que se está registrando, a continuación se muestran cada una de ellas:

Registro de usuarios de tipo DGOSE:

The screenshot shows a web browser window with the address bar displaying <http://www.servicio-social.unam.mx/admn/usuarios/dgose/registro>. The page has a navigation menu at the top with options: Inicio, Estudiantes, Programas, URSS, Instituciones receptoras, and Usuarios. A sidebar on the left contains a tree view with categories: Usuarios, DGOSE, Administración, Registro, URSS, Estudiantes, and Instituciones receptoras. The main content area is titled "Registro de usuarios - DGOSE -" and is divided into two sections: "Información personal" and "Información del sistema".

Información personal

Nombre (s):

Apellido paterno:

Apellido materno:

Cargo:

Correo:

 [Subir imagen](#)

Información del sistema

Nombre de

Nota:

La información para activar la cuenta del usuario será enviada a la dirección de correo electrónico registrada.

De igual forma es necesario presentar el formato: [responsiva_usuarios_dgose.docx](#), firmado por el interesado y aprobado por su jefe de departamento.

Universidad Nacional Autónoma de México
Dirección General de Orientación y Servicio Educativos
Sub Dirección de Servicio Social

Ilustración 53 Registro de usuarios DGOSE. Fuente: elaboración propia.

Registro de usuarios del tipo Unidades Responsables de Servicio Social (URSS):

SIASS - Usuarios
 http://www.servicio-social.unam.mx/admin/usuarios/urss/registro

Inicio Estudiantes Programas URSS Instituciones receptoras **Usuarios**

Usuarios
 DGCOSE
URSS
 Administración
 Registro
 Estudiantes
 Instituciones receptoras

Registro de usuarios - URSS -

Información personal

Nombre (s):

Apellido paterno:

Apellido materno:

Cargo:

 [Subir imagen](#)

Información de contacto

Correo:

Teléfono:

Teléfono celular:

Fax:

Entidad Federativa:

Delegación/Municipio:

Calle y número:

Colonia:

Código postal:

Información académica

Facultad / Escuela:

Carrera	Es responsable
Actuaría	<input checked="" type="checkbox"/>
Biología	<input checked="" type="checkbox"/>
Ciencias de la computación	<input checked="" type="checkbox"/>
Ciencias de la tierra	<input checked="" type="checkbox"/>
Física	<input type="checkbox"/>

Información del sistema

Nombre de

Nota:

La información para activar la cuenta del usuario será enviada a la dirección de correo electrónico registrada.

De igual forma es necesario presentar el formato: [responsivo_usuarios_urss.docx](#), firmado por el interesado y aprobado por su jefe de departamento, especificando la tutoría de las carreras de las cuales es responsable.

Universidad Nacional Autónoma de México
 Dirección General de Orientación y Servicio Educativos
 Sud Dirección de Servicio Social

Ilustración 54 Registro de usuarios URSS. Fuente: elaboración propia.

Y por último el registro de usuarios de tipo Institución Receptora (IR):

Inicio	Estudiantes	Programas	URSS	Instituciones receptoras	Usuarios
--------	-------------	-----------	------	--------------------------	-----------------

Usuarios

DEOSE

URSS

Estudiantes

Instituciones receptoras

Administración

Registro

Registro de usuarios - Instituciones Receptoras -

Información personal

Nombre (s):

Apellido paterno:

Apellido materno:

Cargo:

Información de contacto

Correo:

Teléfono:

Fax:

Entidad Federativa:

Delegación/Municipio:

Calle y número:

Colonia:

Código postal:

Información institucional

Tipo de institución: UNAM Sector Público Federal Sector Social

Institución receptora: [Nueva](#)

Dependencia ejecutora: [Nueva](#)

Información del sistema

Nombre de

Terminos y condiciones:

La solicitud de activación de usuario será revisada por la Dirección General de Orientación y Servicios Educativos quien dará un fallo al respecto y notificará vía correo electrónico a la dirección registrada.

De igual forma es necesario enviar el formato: [responsivo_usuarios_ir.docx](#), con membrete de la institución, firmado por el interesado y aprobado por su jefe de departamento, sud director o director. El archivo digital deberá ser enviado a la dirección de correo electrónico siass@unam.mx en un formato .pdf y nombrado con el número de caso que se presentará al concluir el registro.

Acepto los terminos y condiciones planteados.

Acepto No acepto

Universidad Nacional Autónoma de México
Dirección General de Orientación y Servicio Educativos
Sud Dirección de Servicio Social

Ilustración 55 Registro de usuarios IR. Fuente: elaboración propia.

Manual de Usuario – Instituciones Receptoras



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**MANUAL DE USUARIO
INSTITUCIONES Y
DEPENDENCIAS**



SECRETARÍA DE SERVICIOS A LA COMUNIDAD
DIRECCIÓN GENERAL DE ORIENTACIÓN Y SERVICIOS EDUCATIVOS
2015

INTRODUCCIÓN

La Dirección General de Orientación y Servicios Educativos (DGOSE) tiene como una de sus funciones relevantes la coordinación del Servicio Social Universitario, en conjunto con las Unidades Responsables de Servicio Social de Facultades, Escuelas y Centros de Enseñanza de Nivel Superior.

El Servicio Social es una actividad esencial que consolida la formación profesional y fomenta en el estudiante una conciencia de solidaridad con la comunidad, la sociedad y el país. Por ello es importante dar a conocer el interés de la UNAM por resignificar y redimensionar la prestación del servicio social; proyectar a las instituciones y dependencias el objetivo del servicio social universitario, tomando en consideración los tres ámbitos que lo componen: Formativo, Social y Retributivo; establecer con las dependencias públicas y sociales un compromiso para reorganizar programas específicos de acuerdo a los Ejes de Acción, con el propósito de completar la formación profesional del estudiante universitario y, al mismo tiempo, cumplir con las metas sociales, de servicio y de retribución de las instituciones y dependencias.

El servicio social universitario es una parte del proceso formativo profesional, que se cumple mediante las acciones que realizan los estudiantes de las carreras técnicas y profesionales, de manera temporal y obligatoria, tendientes a la retribución social en beneficio o interés de las comunidades, a través de la aplicación de los conocimientos, habilidades, actitudes y valores adquiridos.

Para la UNAM, el servicio social es una práctica académica de beneficio social y no un recurso humano calificado que puede ser utilizado para cubrir necesidades o carencias que a las instancias solicitantes no les es posible atender con recursos propios.

Por lo anterior, el presente manual tiene como finalidad apoyar al usuario en el manejo del Sistema de Información Automatizada de Servicio Social (SIASS) describiendo puntualmente todas sus funciones.

ACCESO A LA PÁGINA PRINCIPAL DEL SIASS

Para ingresar a la página del SIASS, se deberá seguir los siguientes pasos:

- 1) Abrir el Navegador de Internet (Internet Explorer, Google Chrome, Mozilla Firefox, etc.)
- 2) Ingresar la dirección <http://www.siaass.unam.mx/> que corresponde al SIASS Web 2016, el nuevo sistema de información del Servicio Social Universitario de la DGOSE.
- 3) A continuación se mostrará la página principal del SIASS.

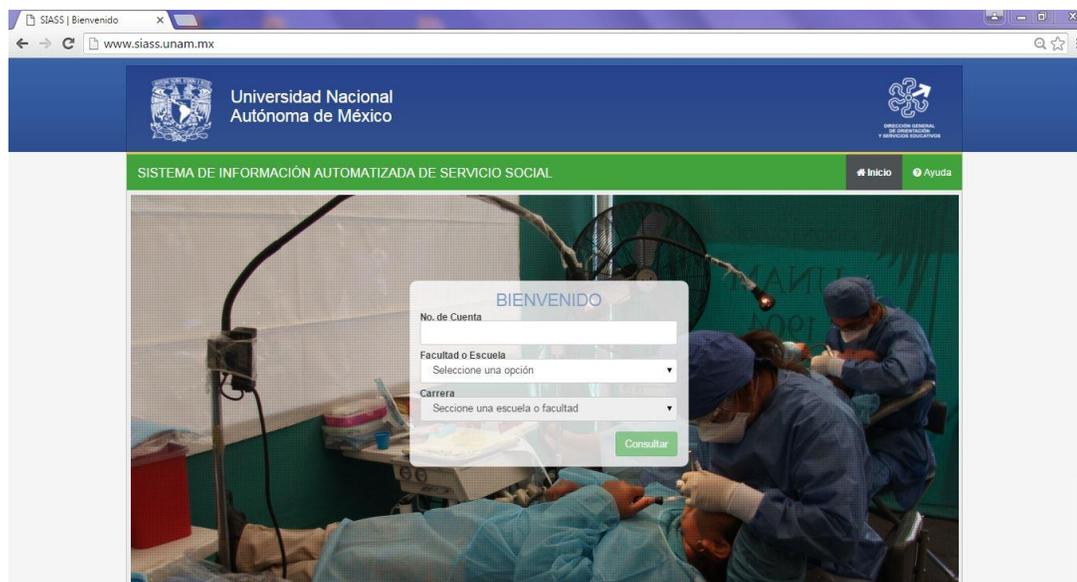


Imagen 1: Página de Inicio 1

El sistema cuenta con tres botones ubicados en la parte superior derecha:



Consulta: este botón llevará directamente a la página principal del sistema, es decir, el sitio de consulta de programas aprobados para las diferentes facultades y carreras.



Administración: mediante este botón se accederá al portal de administración para usuarios URSS e Instituciones Receptoras. En este portal es donde se podrá iniciar sesión para realizar los procesos correspondientes a cada tipo de usuario.



Ayuda: aquí se encontrará material de apoyo para el uso del sistema así como todos los lineamientos vigentes para el registro de programas.

MÓDULOS DEL SISTEMA SIASS

Al ingresar al sitio de administración, se mostrará la siguiente pantalla



Imagen 2: Módulos del SIASS 1

I. Gestión Institucional

En la pantalla se visualizará un menú del lado izquierdo con los siguientes iconos:

- Iniciar sesión.
- Registro de nuevos usuarios.
- Ingresar folio

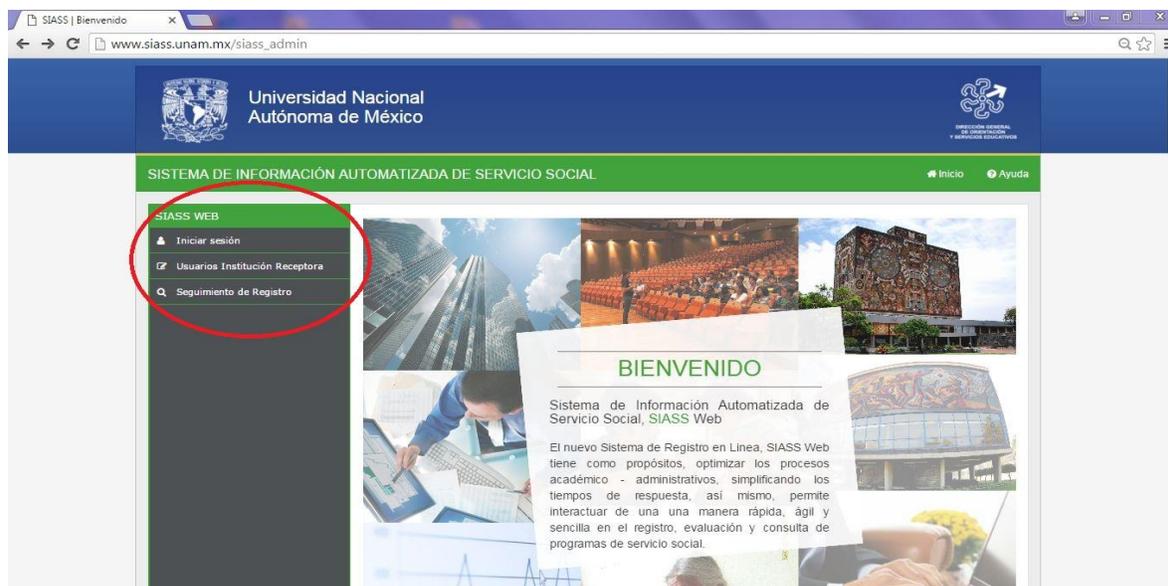


Imagen 3: Menú, Página de Administración 1

Registro de nuevos usuarios

Para ingresar al SIASS deberá realizar su registro como institución receptora. Dicha solicitud será analizada por la DGOSE, quien notificará vía correo electrónico el resultado correspondiente.

1. En el menú (parte izquierda de la pantalla) se seleccionará la opción **“Registro de nuevos usuarios”**.

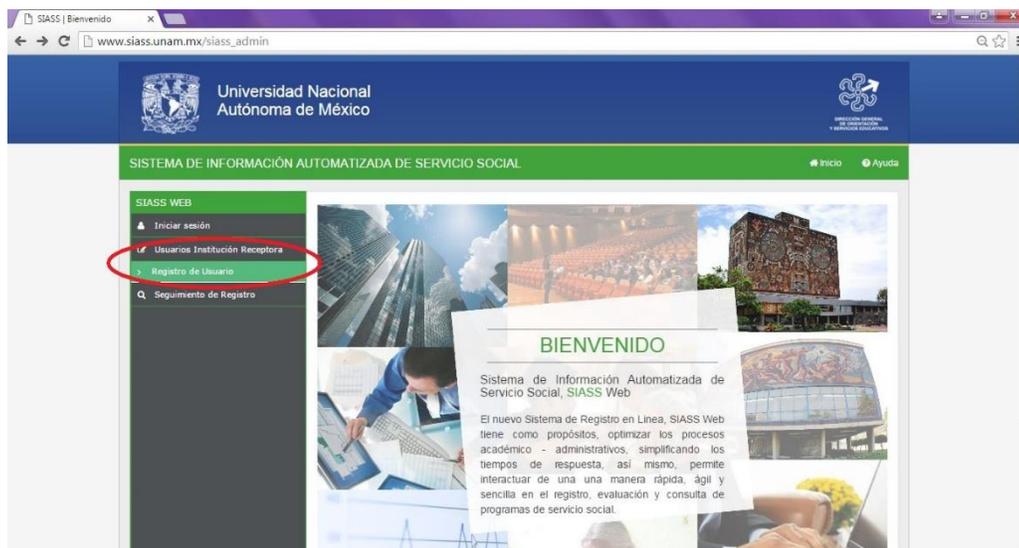


Imagen 4: Opción Registro de Usuarios 1

2. El sistema mostrará la pantalla siguiente en la cual pedirá información acerca de la institución.
 - a. Primero se deberá seleccionar el sector al que pertenece la institución receptora (público o social).
 - b. Después se seleccionará el nombre de la institución de un listado.

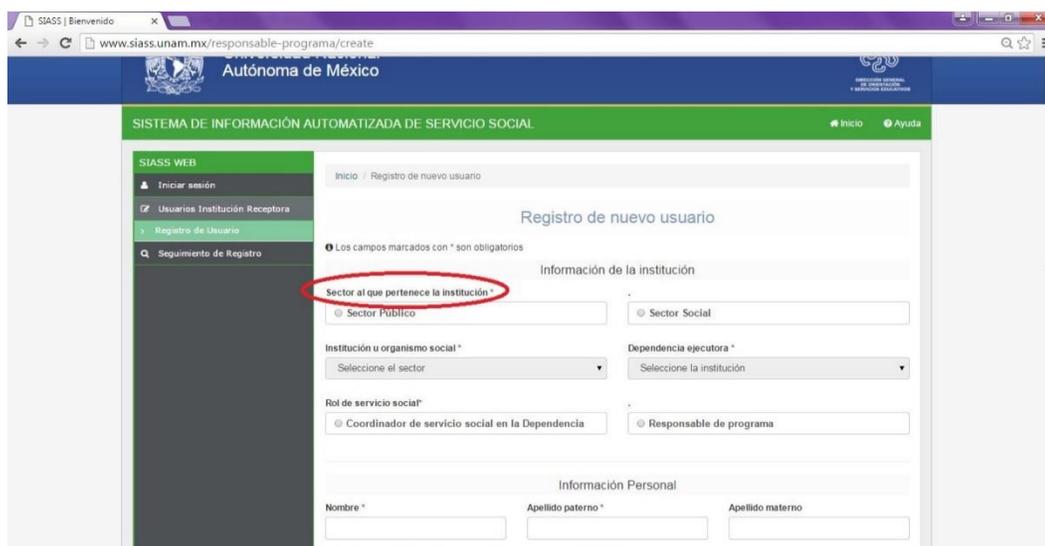


Imagen 5: Sector de la Institución 1

3. En caso de que la institución no se encuentre en la lista, se deberá buscar al final de la misma y se seleccionara la opción **“Crear una nueva institución u organismo social”**.

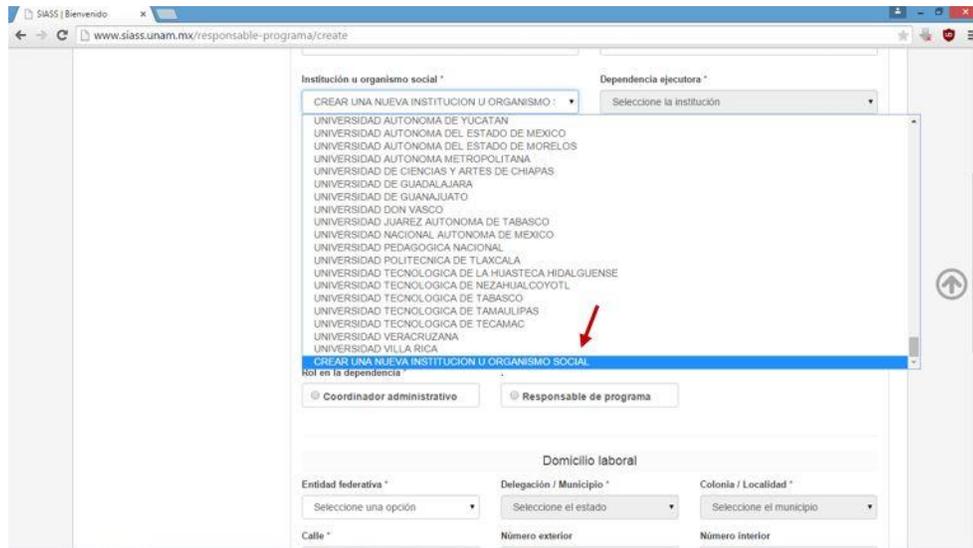


Imagen 6: Crear una nueva Institución 1

4. Después, se ingresará el nombre completo de la nueva institución, así como también el nombre de la nueva dependencia (ambas sin abreviaturas).

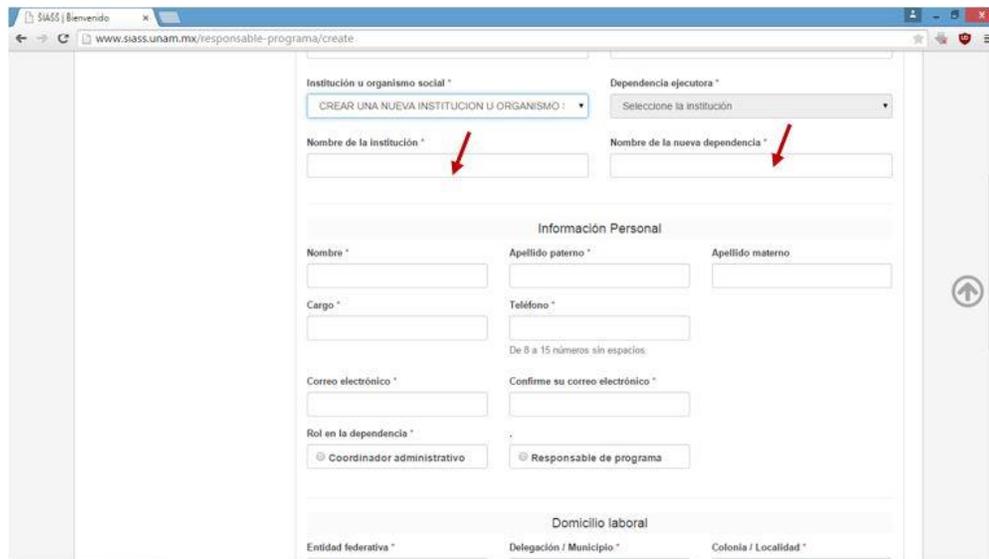


Imagen 7: Datos generales Institución 1

5. Una vez hecho lo anterior, se llenarán los campos solicitados con la información correspondiente de la persona responsable de la institución receptora.

Imagen 8: Información del responsable 1

- Finalmente se deberá agregar el archivo de registro de firmas en formato PDF o imagen en formato JPG, para eso se dará clic en **“Seleccionar archivo”**. Una vez hecho lo anterior se dará clic en **“Enviar formato”**, para finalizar el registro de una nueva institución receptora.

Imagen 9: Archivo de firmas 1

- Una vez enviada la solicitud, se mostrará una confirmación de que la información fue recibida de manera correcta. Además se recibirá un correo con el número de folio para dar seguimiento al proceso de validación de la nueva institución receptora hecho por **DGOSE**.

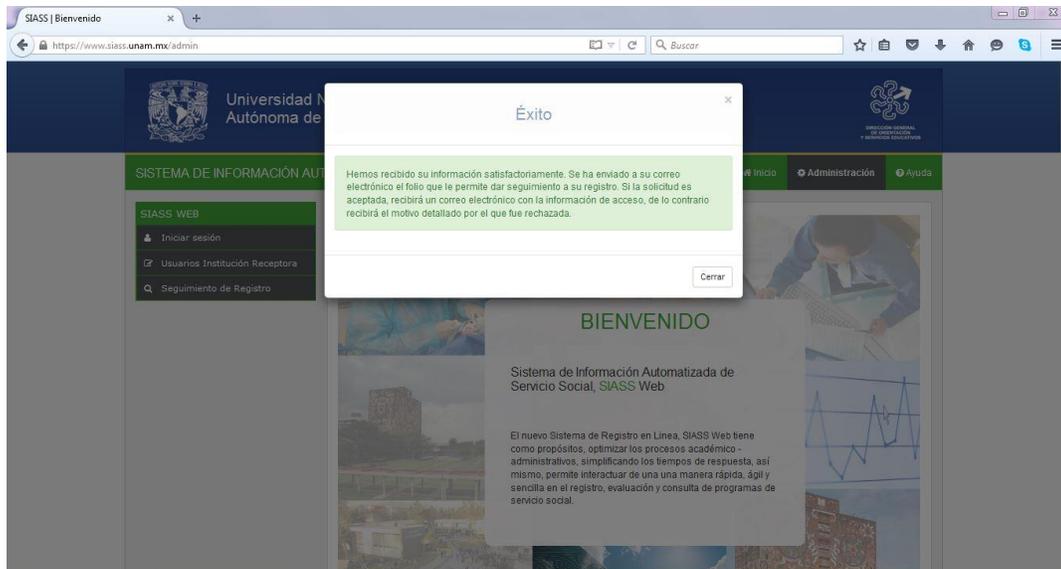


Imagen 10: Confirmación de registro 1

8. Para poder visualizar si la institución ha sido validada o no, se ingresará en la opción del menú **“Ingresar folio”** y se usará el folio recibido vía correo electrónico.

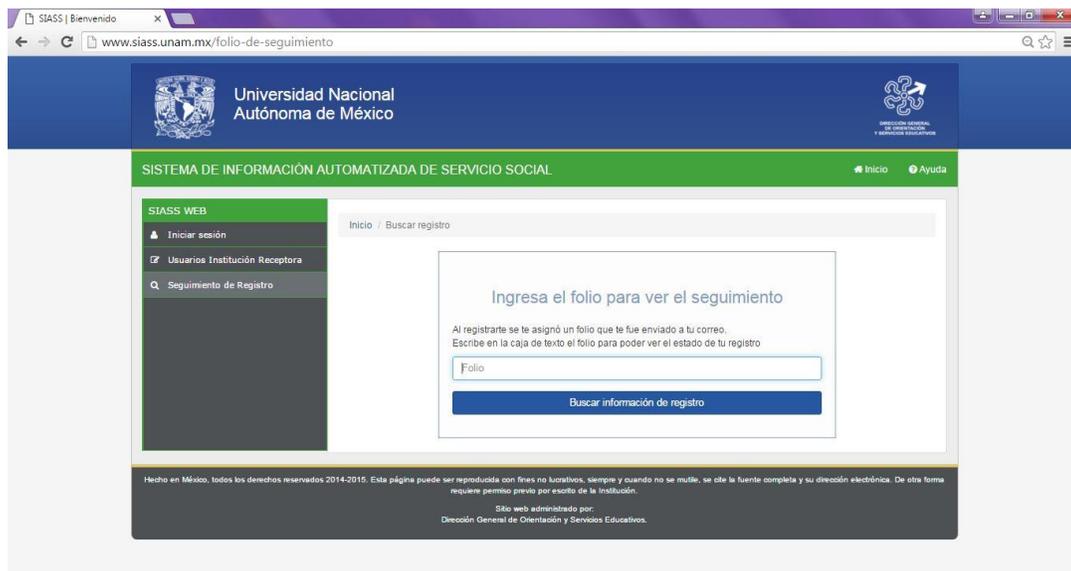


Imagen 11: Número de folio 1

9. En la pantalla se indicará el estado en que se encuentra la institución dentro del proceso de validación hecho por DGOSE. Se tendrán las siguientes opciones para el estado de la institución:
 - a. En revisión:
 - b. Aprobado: la institución ha sido aprobada y puede acceder al SIASS.
 - c. No aprobado: la institución no fue aprobada y recibirá un correo electrónico con los motivos.

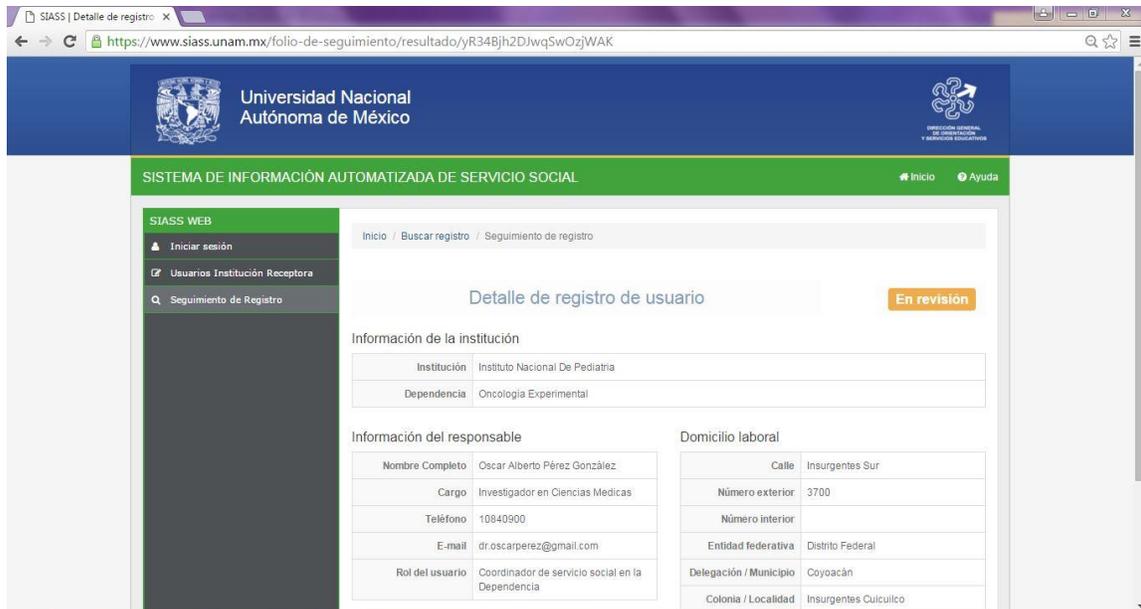


Imagen 12: Estado del programa 1

Si la solicitud fue aprobada, se enviará un correo electrónico al responsable de la institución con el nombre de usuario y la contraseña.

El Coordinador Administrativo estará obligado a mantener la confidencialidad de su cuenta y dar buen uso a la misma.

Ingreso al sistema

Una vez que la solicitud para registrar una nueva institución receptora haya sido revisada y aprobada por DGOSE, se podrá tener acceso al SIASS mediante el usuario y la contraseña proporcionados al correo electrónico del responsable de la institución.

Para poder ingresar como instituciones receptoras al SIASS:

- Se seleccionará la opción “**Iniciar sesión**”



Imagen 13: Iniciar Sesión 1

- En la siguiente pantalla se ingresará el usuario y la contraseña proporcionados en los campos correspondientes.

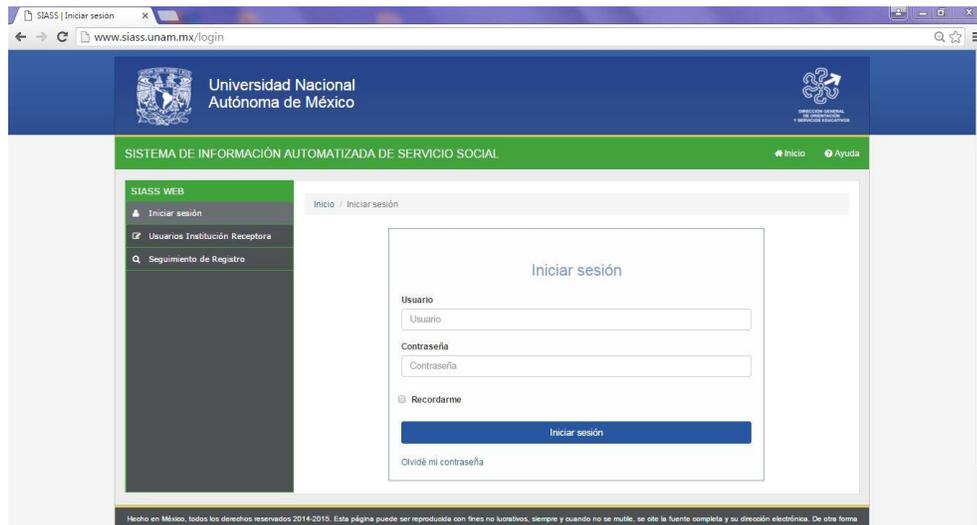


Imagen 14: Formulario inicio sesión 1

Una vez dentro del SIASS se podrá visualizar una ventana como la que se muestra a continuación, en la cual se tendrán las siguientes opciones.

- Cambiar contraseña: se podrá modificar la contraseña proporcionada por la DGOSE.
- Ver firmas: se visualizará el archivo de firmas.
- Programas de servicio social:
 - Listado de programas : se mostrará el listado de todos los programas de servicio social de la institución receptora. (si se ha entrado por primera vez al SIASS, no habrá ningún programa registrado).
 - Crear programa: se podrán crear nuevos programas de servicio social para la institución.
- Cerrar sesión: se cerrará la sesión actual de manera segura.

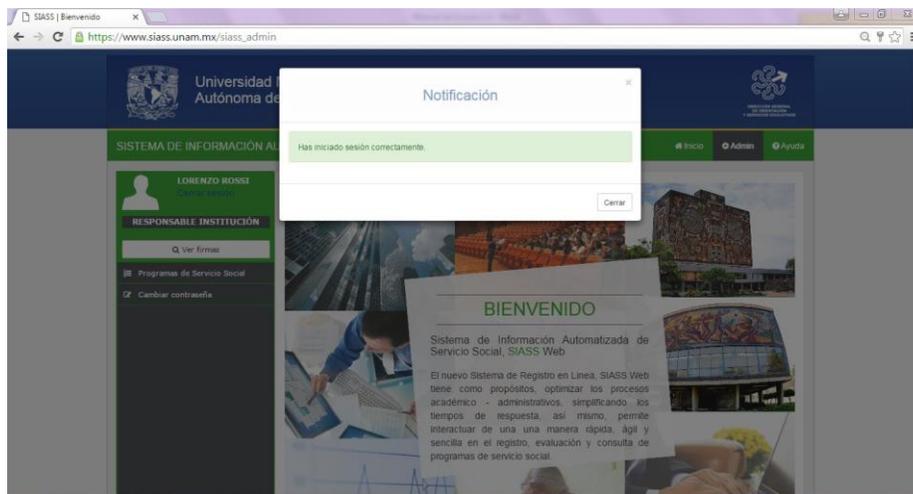


Imagen 15 - Confirmación inicio sesión



Imagen 16: sesión iniciada 1

Cambiar contraseña de usuario

Para poder modificar la contraseña proporcionada por la DGOSE se deberán realizar las acciones siguientes:

- Una vez que se ha accedido al SIASS por medio del usuario y la contraseña, seleccionar la opción “**Cambiar contraseña**”.
- Una vez realizado lo anterior, se ingresará la contraseña original y la nueva contraseña así como la confirmación de la misma.

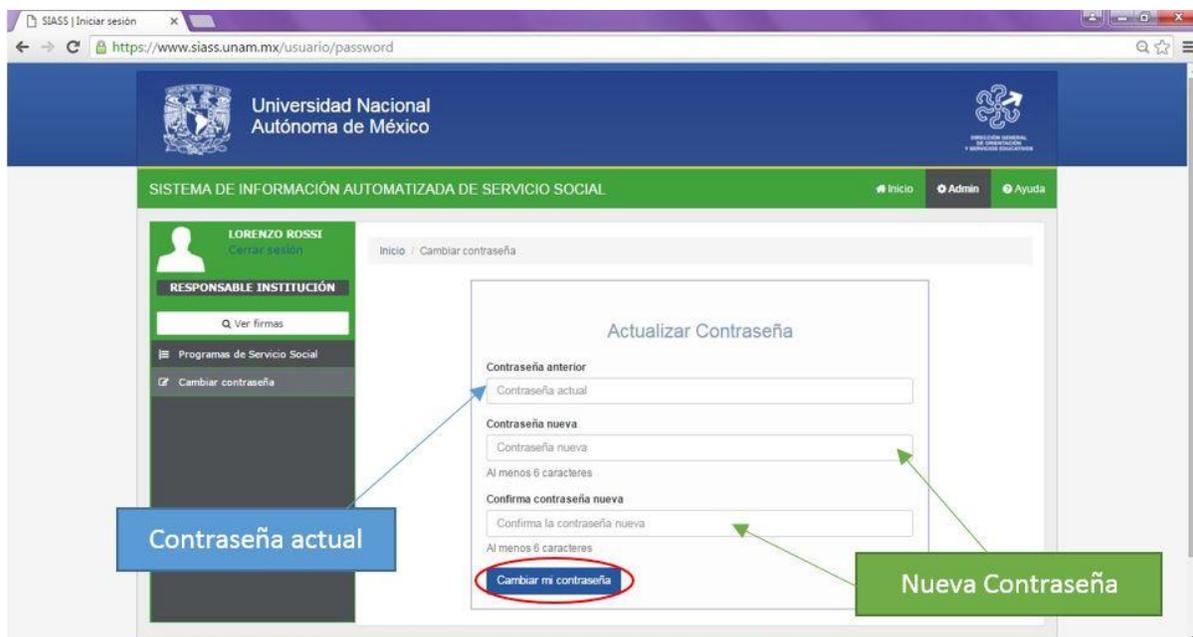


Imagen 17: cambiar contraseña 1

- Para terminar solo se dará clic en el botón azul “**Cambiar mi contraseña**” y se habrá actualizado la contraseña anterior.

Ver documento de firmas

Si se quiere ver el documento anexo de firmas solo se deberá dar clic en el botón **“Ver firmas”**, lo cual mostrara una pantalla como la siguiente.

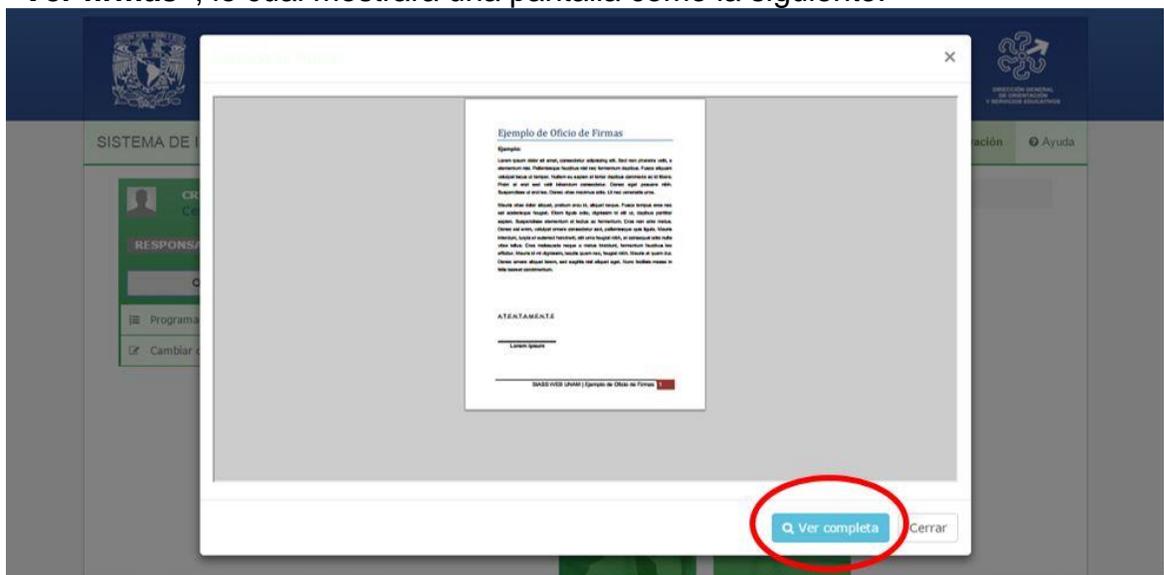


Imagen 18: ver documento de firmas 1

Si se da clic en el botón **“Ver completa”** se podrá ver el documento en una nueva pestaña en el navegador web o se descargará al ordenador (dependiendo del navegador que se esté utilizando).

Alta de nuevos programas

Se ingresarán nuevos programas cuando:

- a) Se trate de instituciones que soliciten por primera vez prestadores de servicios social a la UNAM.

- a) Se trate de instituciones con antecedentes de registro ante la UNAM, que tengan programas vigentes, pero que requieran registrar programas nuevos o distintos.

Para registrar nuevos programas en el SIASS Web:

1. Ingresar a la página principal del SIASS Web con su nombre de usuario y contraseña.
2. Dar clic en el botón **“Iniciar sesión”**.
3. A continuación en la siguiente pantalla se dará clic en el botón **“Programas de Servicio Social”** y después en la opción **“Crear programa”**.



Imagen 19: Menú de programas 1

- En la siguiente pantalla se mostrará la información de la dependencia y el responsable, se deberá completar el resto de la información que se solicita en la página.

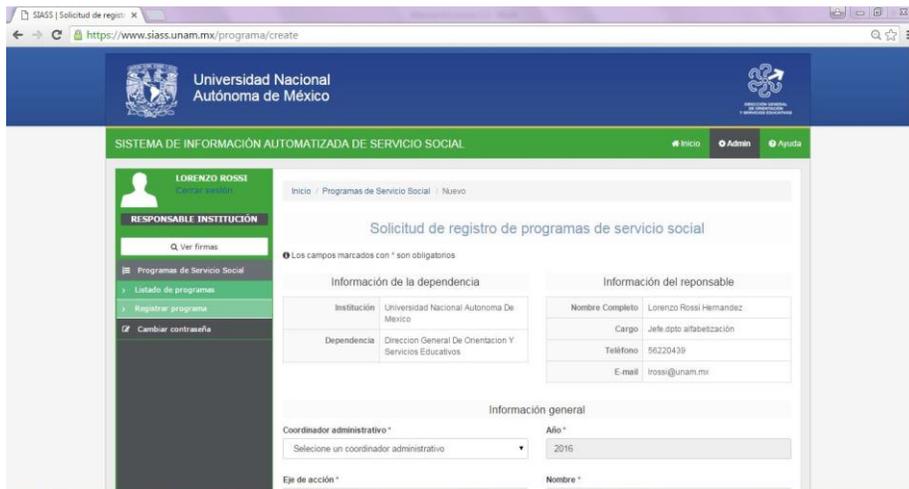


Imagen 20: Información dependencia 1

- En la sección “**Información general**” se deberá elegir un coordinador administrativo, así como también un eje temático y adicionalmente se pondrá el nombre del programa de servicio social como se muestra en la imagen siguiente.

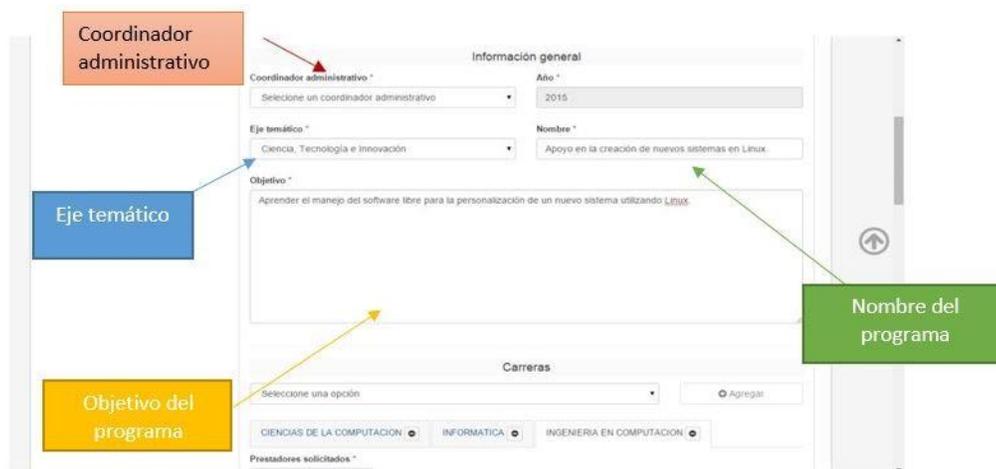


Imagen 21: Información general programa 1

- En la sección **“Carreras”** se deberá dar clic en la flecha **“Seleccione una opción”** para desplegar el catálogo de todas las carreras, una vez encontrada la carrera, seleccionarla y dar clic en el botón **“Agregar”**.

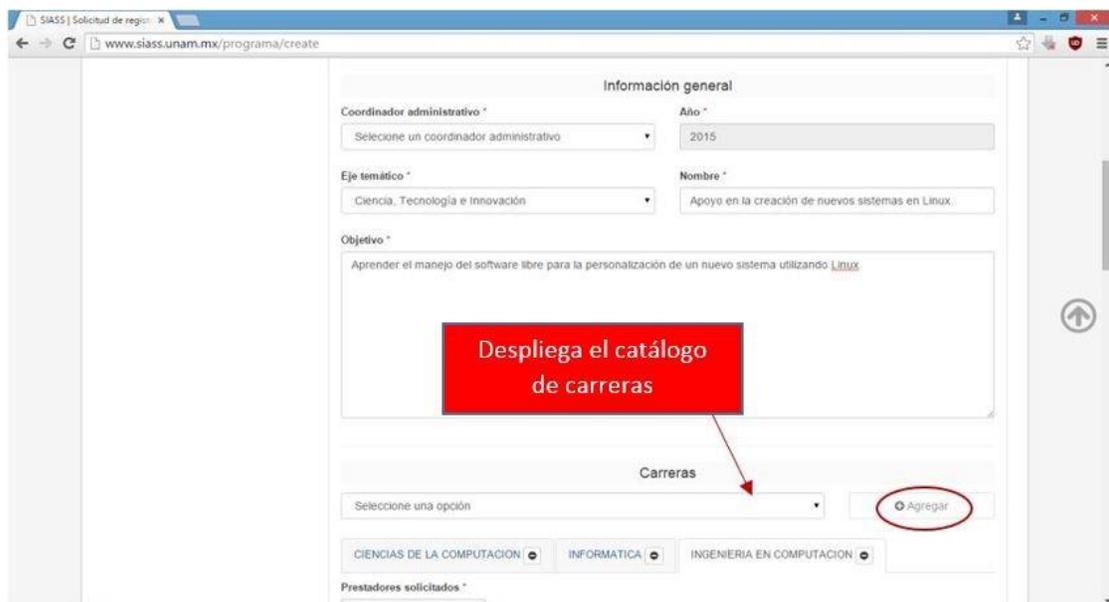


Imagen 22: Información Carreras 1

- Al seleccionar la carrera, se despliega una pantalla que solicita el número de prestadores de servicio social requerido y las actividades a desarrollar considerando el perfil profesional de los mismos. Deberá ingresar 5 actividades como mínimo por cada carrera.



Imagen 23: Prestadores y actividades 1

8. Repetir los pasos 6 y 7 para agregar todas las carreras solicitadas para el programa de servicio social.
9. En la sección “**Ubicación geográfica del programa**” se elegirá un botón de acuerdo al lugar de desempeño de actividades del prestador de servicio social, también se ingresará la dirección completa.

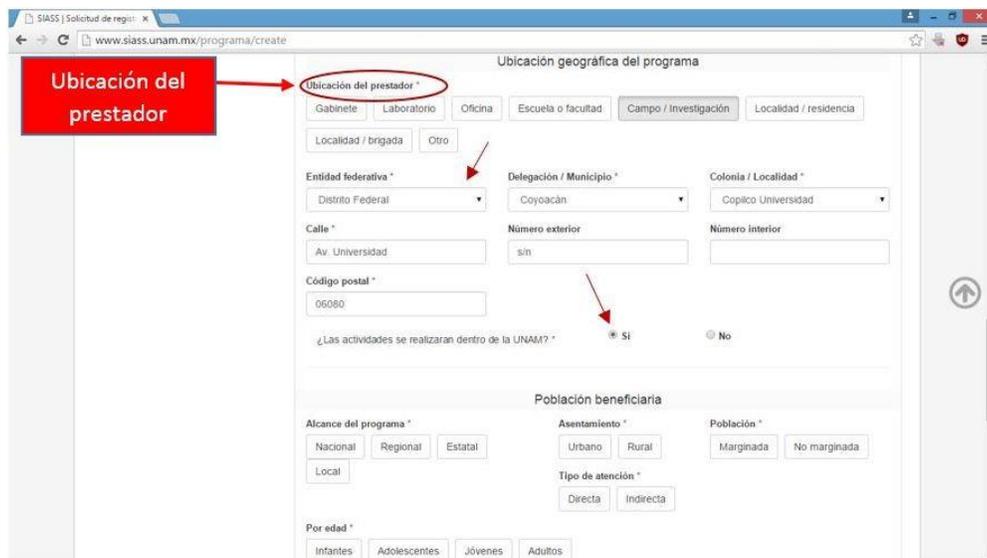


Imagen 24: ubicación geográfica 1

10. En la sección “**Población beneficiada**” se elegirá las opciones de acuerdo con la población que se beneficiará con el programa de servicio social.

Imagen 25: población beneficiaria 1

11. Para la sección “**Apoyo a estudiantes**” se deberá seleccionar las opciones correspondientes con los apoyos que el programa brinde a los prestadores de servicio social. En la sección de “**Asistencia**” tendrán que ir marcados los días y el horario de asistencia de los prestadores. Al terminar se dará clic en el botón “**Enviar formato**” para terminar el registro del nuevo programa.

Imagen 26: Apoyo a estudiante 1

12. Se mostrará la siguiente notificación, la cual, indicará que el programa ha sido registrado correctamente y pasará a revisión por cada responsable de URSS de las carreras elegidas para ese programa.

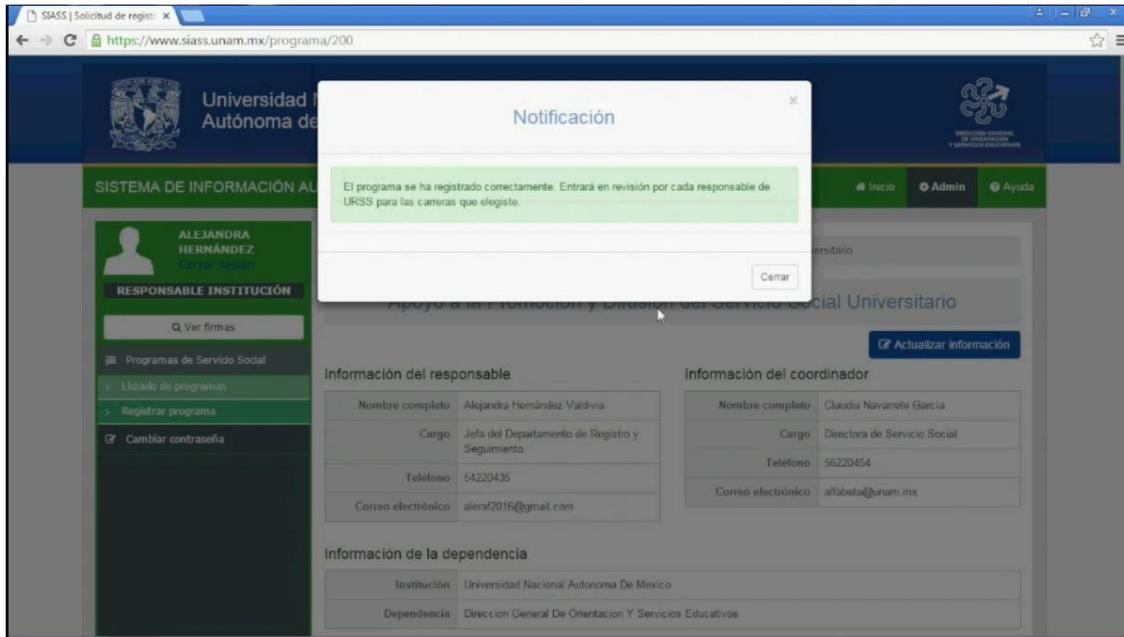


Imagen 27: Programa registró exitoso 1

13. Todos los programas dados de alta serán revisados por la DGOSE, en caso de que el planteamiento sea correcto, se le asignará clave de registro y pasará a la evaluación respectiva.
14. Si existen observaciones al programa, la DGOSE notificará vía correo electrónico una solicitud de replanteamiento del mismo, en caso contrario, se deberá verificar el estado de la evaluación del programa.

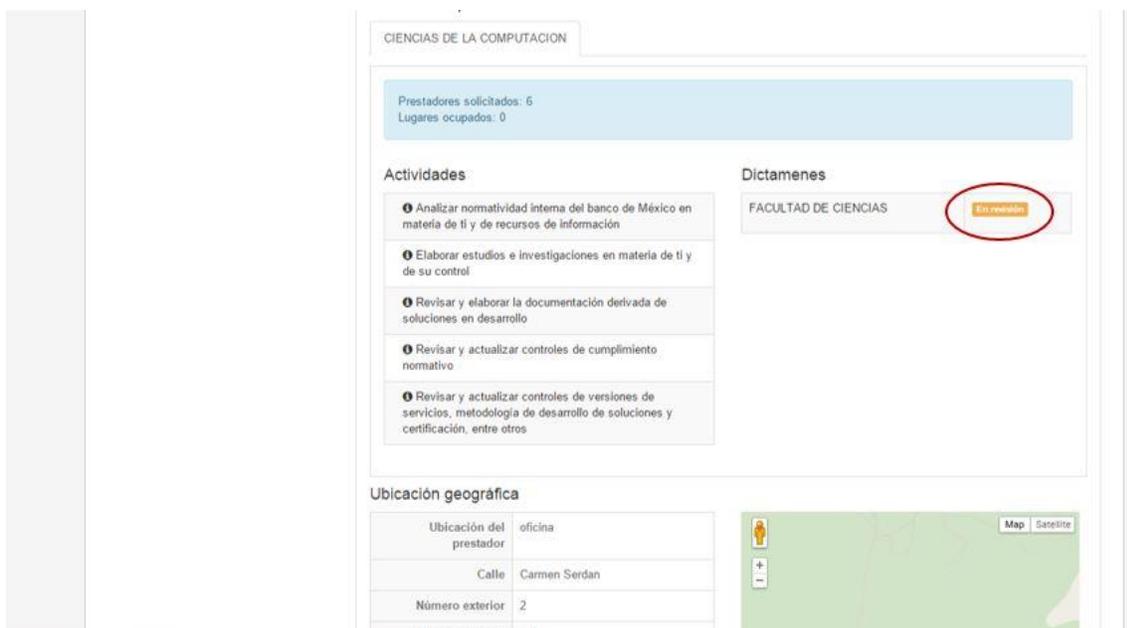


Imagen 28: estado de programas 1

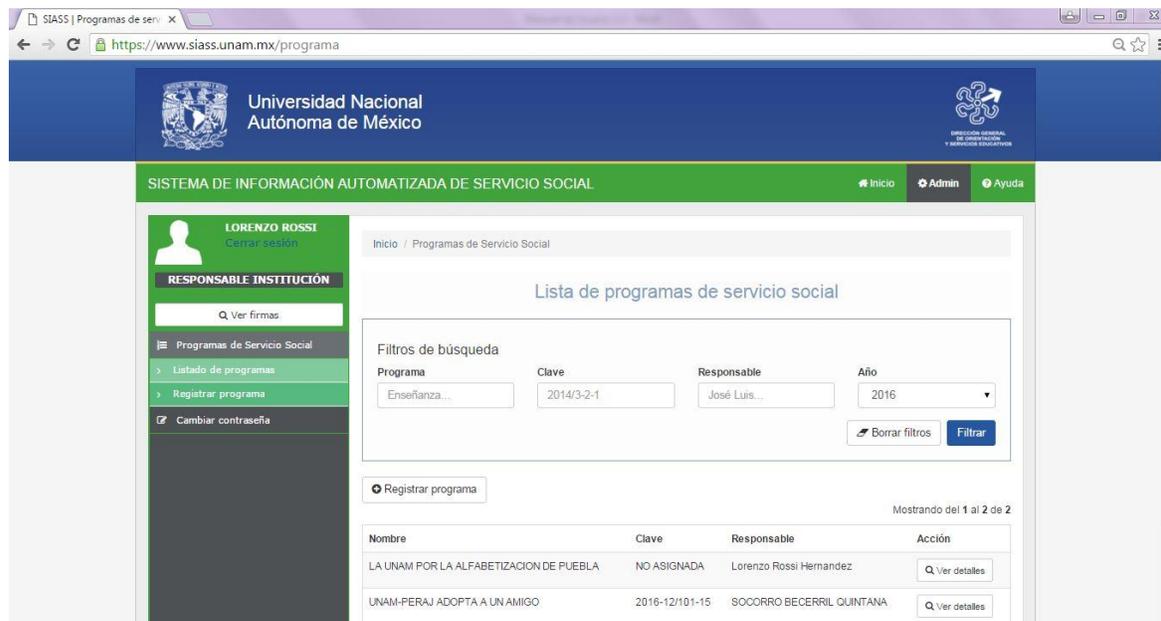
Modificación de programas

La modificación de programas se realizará cuando:

- Las instituciones y/o dependencias modifiquen las secciones de sus programas, siempre y cuando la DGOSE no les haya asignado clave.
- Cuando el programa cuenta ya con una clave asignada, para el periodo en curso es posible modificar información de tipo general así como incorporar carreras con sus respectivas actividades, para lo cual es necesario hacer la solicitud mediante oficio a la DGOSE.

Para la modificación de programas sin clave debe:

- Seleccionar “**Listado de programas**”, buscar el programa a modificar y dar clic en el botón “**Ver detalles**”.



The screenshot shows the SIASS web application interface. The header includes the Universidad Nacional Autónoma de México logo and the text 'SISTEMA DE INFORMACIÓN AUTOMATIZADA DE SERVICIO SOCIAL'. The user profile 'LORENZO ROSSI' is visible. The main content area is titled 'Lista de programas de servicio social' and features search filters for 'Programa', 'Clave', 'Responsable', and 'Año'. Below the filters is a table with two rows of program data.

Nombre	Clave	Responsable	Acción
LA UNAM POR LA ALFABETIZACION DE PUEBLA	NO ASIGNADA	Lorenzo Rossi Hernandez	Ver detalles
UNAM-PERAJ ADOPTA A UN AMIGO	2016-12/101-15	SOCORRO BECERRIL QUINTANA	Ver detalles

Imagen 29: Listado de programas 1

- Dar clic en el botón “**Actualizar información**” permitirá actualizar la información del programa.

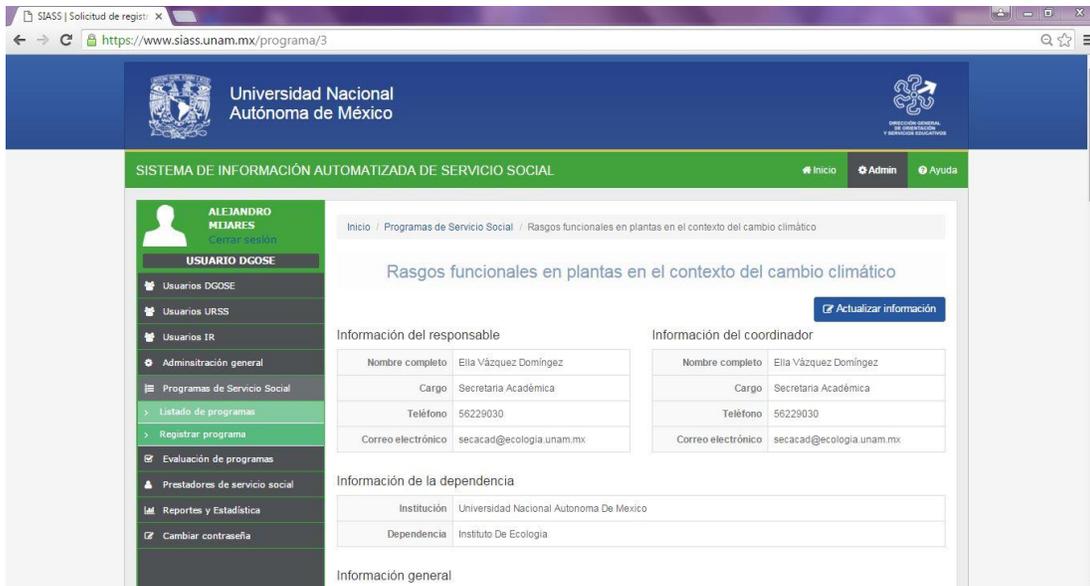


Imagen 30: Actualiza programas 1

- Una vez realizados los cambios se deberá dar clic en el botón **“Actualizar programa”**, ubicado en la parte inferior y el sistema confirmará que los datos han sido actualizados.

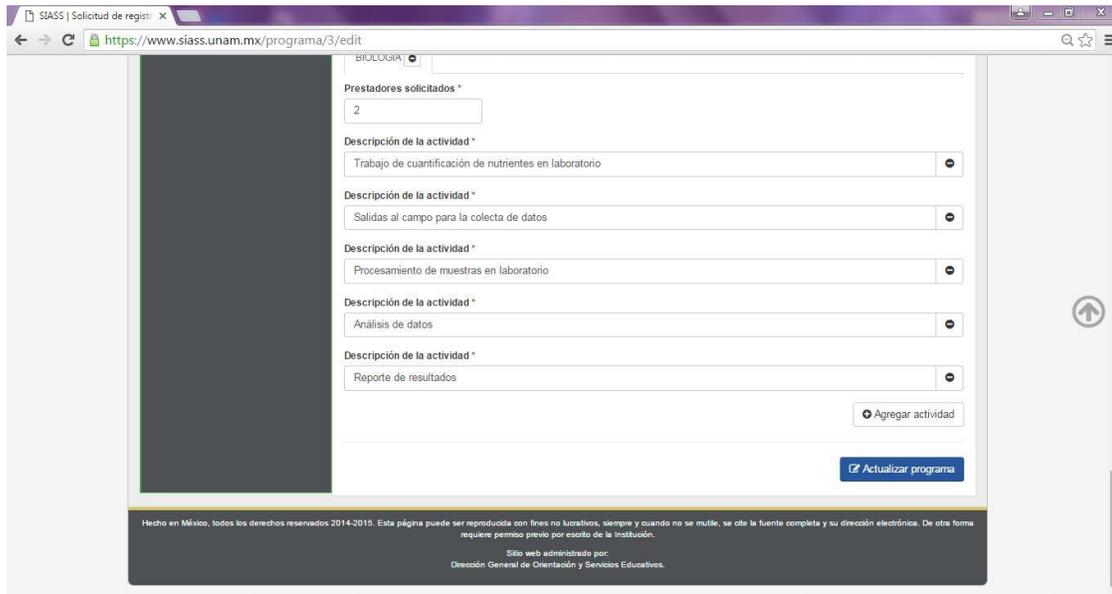


Imagen 31: actualiza programas 2 1

- Una vez realizados todos los cambios se deberá salir de SIASS utilizando el botón **“Cerrar sesión”**.

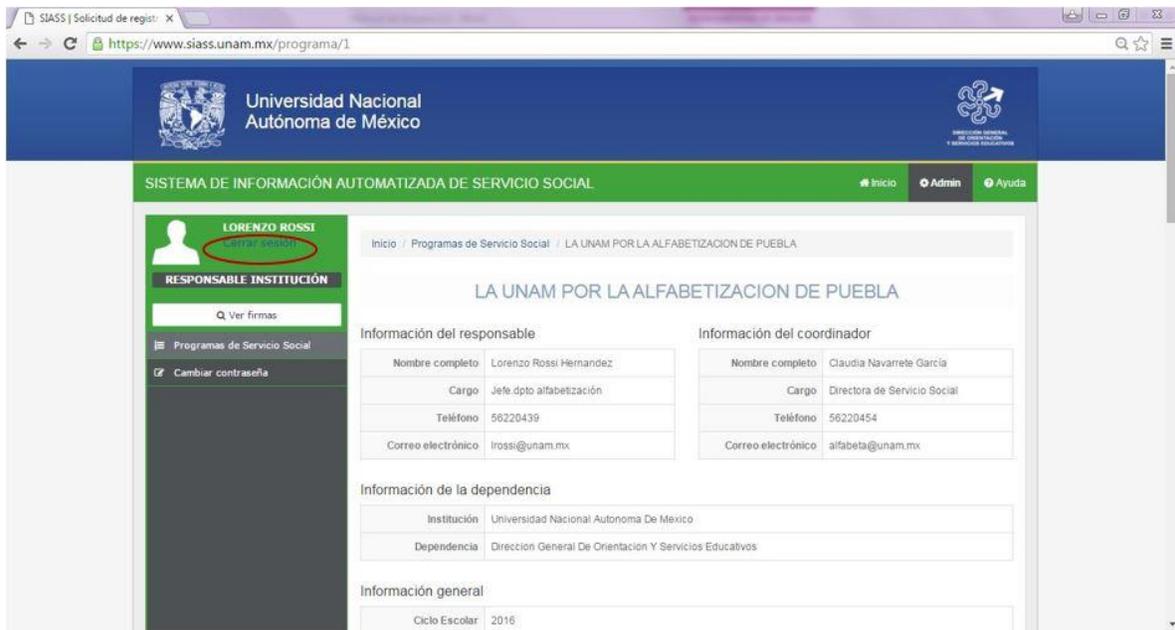


Imagen 32: cerrar sesión 1

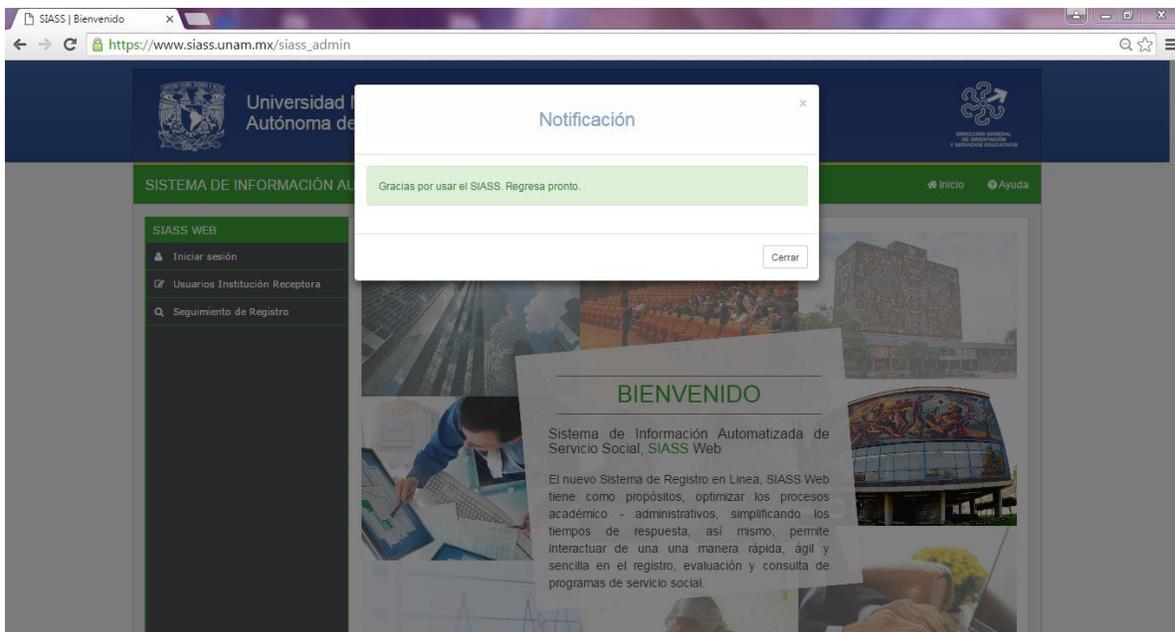


Imagen 33 - Confirmación sesión cerrada 1