

UNIVERSIDAD POLITÉCNICA DE PUEBLA
Ingeniería en Informática



**Proyecto de Estancia Práctica en
Desarrollador en Sistemas de Software y
Administrador de Redes**

“Sistema de Gestión para Responsivas de Uso de TIC”

Área temática del CONACYT: VII
Ingenierías y tecnologías

Presenta:

Lourdes Nataly Rojas Hernández

Asesor técnico

M.C. Víctor Manuel Cortés Castillo

Asesor académico

M.C. Rebeca Rodríguez Huesca

Juan C. Bonilla, Puebla, México.

28-Agosto-2019

Resumen

Este proyecto consiste en el desarrollo de un sistema de 'gestión para responsivas del uso de TIC', que usará la Administración General de Cómputo (AGC) del Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), brindando una mejor gestión sobre el control de los datos que se generan por cada: Estudiante, Servicio social, Visitante, Proveedor y Trabajador del INAOE. Este documento responde a las preguntas de: ¿Por qué se realizó el sistema?, ¿Qué debe de hacer?, ¿En qué metodología está basado?, ¿Qué objetivos tiene?, ¿Qué herramientas se utilizaron?, ¿Cómo fue el desarrollo? Y ¿Qué resultados se obtuvieron?.

El desarrollo del sistema se describe de manera detalla, principalmente para que cualquier persona que lea el documento pueda entender todas sus fases.

Índice

	Pag.
1. Introducción	4
1.1 Descripción del problema o necesidad.	4
1.2 Justificación	4
1.3 Objetivo General y Específicos	5
2. Metodología y herramientas	6
2.1 Metodología a seguir	6
2.2 Herramientas tecnológicas utilizadas	9
2.3 Razones por la cual se eligió la metodología	12
Prototipo	
3. Resultados	13
3.1 Investigación Preliminar	13
3.2 Especificaciones de requerimientos y prototipado	13
3.3 Diseño Técnico	17
3.4 Programación y Prueba	23
3.5 Operación	26
4. Conclusiones y recomendaciones	27
5. Referencias bibliográficas	28

1. Introducción

Este capítulo contiene los objetivos que el sistema debe cumplir.

1.1 Descripción del problema o necesidad

Actualmente en la Administración General de Cómputo (AGC) del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) no se cuenta con una herramienta administrativa para la gestión de las responsivas de uso de TIC que la AGC requiere para permitir el uso de la infraestructura de red por parte de los usuarios. La AGC actualmente maneja estos documentos en papel y hace búsquedas manuales de los usuarios para diferentes fines. Por otro lado, no se tiene un control automatizado de los usuarios (tales como alumnos, trabajadores, prestadores de servicio social, becarios, proveedores y visitantes) que utilizan los servicios de cómputo que otorga el Instituto, así mismo, no se cuenta con una estadística precisa y al momento de la prestación de servicios.

1.2 Justificación

Este proyecto tiene la finalidad de proveer un sistema de TI que automatice el proceso que actualmente se lleva a mano por parte del personal de la AGC, que permita almacenar una copia digital de las responsivas de uso de TIC, así mismo otra funcionalidad propuesta es la búsqueda de las personas antes mencionadas que llenaron una responsiva para el uso de TIC, finalmente se mostrara una estadística de los alumnos activos e inactivos por medio de una gráfica. Esto será de mucha ayuda para el área de Administración General de Cómputo (AGC) ya que se podrá reducir el tiempo de búsquedas de usuarios y será de manera más precisa. Por otro lado, a través de este desarrollo demostramos las competencias adquiridas durante el proceso de formación como ingeniero en Informática. Cabe resaltar que se obtendrá un mejoramiento en un proceso cotidiano de la Administración General de Cómputo (AGC).

1.3 Objetivo General y Específicos

Objetivo general: Desarrollar un Sistema de Gestión para Responsivas de Uso de TIC para la Administración General de Cómputo que automatice el proceso que actualmente se tiene.

Objetivos específicos:

- El sistema deberá permitir el acceso al módulo de administración solamente al personal autorizado.
- El sistema deberá permitir el registro de nuevos usuarios administradores de manera sencilla.
- El sistema deberá permitir el registro de nuevos usuarios de red de manera sencilla.
- El sistema deberá permitir la modificación de status Activo a Inactivo por cada registro.

2. Metodología y herramientas

En este segundo capítulo se describe la metodología a utilizar para la realización del sistema, las características, así como las ventajas y desventajas de las herramientas utilizadas.

2.1 Metodología a seguir

El sistema de gestión para responsivas de uso de TIC se desarrollará bajo los ciclos de vida de la metodología de "Prototipo" (Figura 1), ya que el prototipo debe ser construido en poco tiempo, usando los programas adecuados ya que no se deben utilizar muchos recursos. La metodología de Prototipo permite que todo el sistema, o algunos de sus partes, se construyan rápidamente para comprender con facilidad y aclarar ciertos aspectos en los que se aseguren que el desarrollador, el usuario, el cliente estén de acuerdo en lo que se necesita, así como también la solución que se propone para dicha necesidad y de esta forma minimizar el riesgo y la incertidumbre en el desarrollo [1].

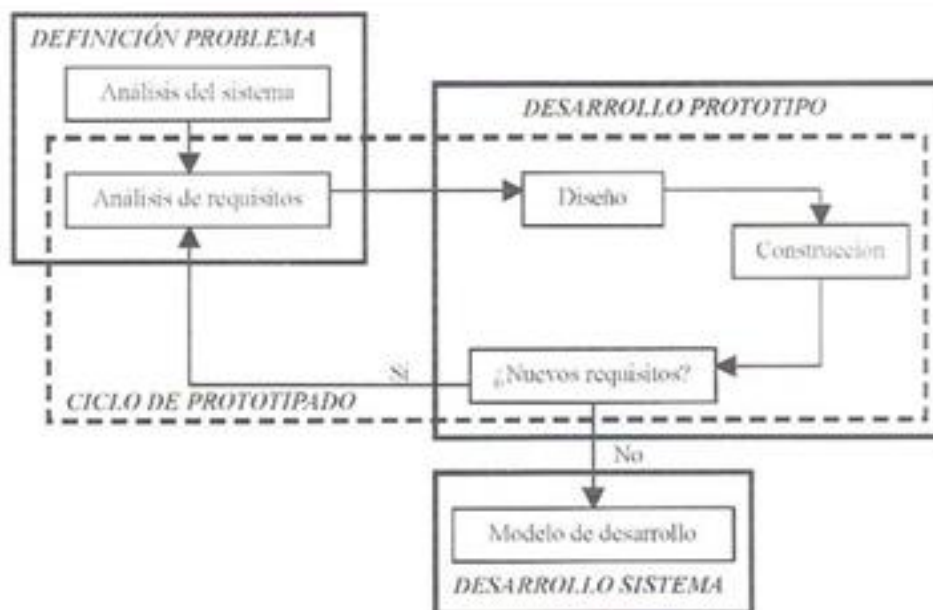


Figura1.Ciclo de vida de la Metodología Prototipo.

2.1.1 Etapas de la metodología en el proyecto.

La gestión del proyecto comprende de las siguientes etapas [2] como se muestra en la figura 2.

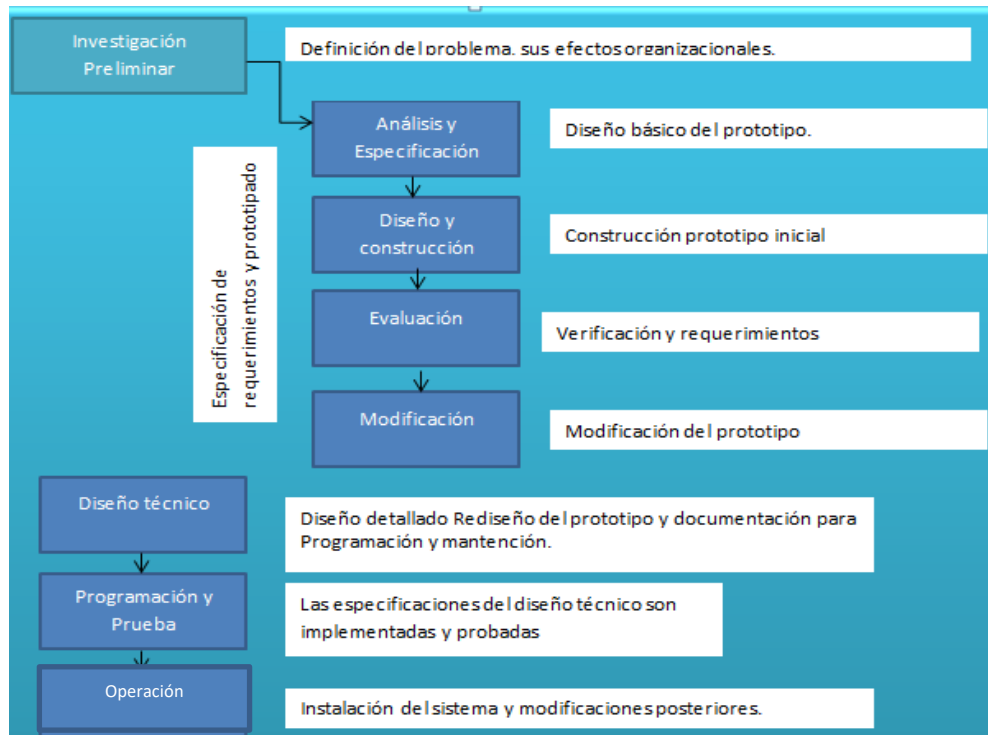


Figura 2. Etapas de la metodología Prototipo.

1.- Investigación preliminar

En esta etapa es determinar el problema y su ámbito, la importancia que tendrán sobre la organización, identificar una idea general de la solución.

2.- Especificación de requerimientos y prototipado.

Esta es la fase más importante de todo el ciclo de vida del método de prototipos, el objetivo en esta fase es determinar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto que se está deseando implementar.

Aquí el desarrollador interactúa con el usuario y sus necesidades mediante la construcción, demostración y retroalimentaciones del prototipo

2.1.-Análisis y Especificación.

Esta etapa es un proceso que busca aproximar las visiones del usuario y del desarrollador mediante sucesivas iteraciones.

Para la definición de los requerimientos tenemos cinco etapas entre dos de las cuales se establece un ciclo interactivo.

2.2.- Diseño y construcción.

Lo que se consigue en esta fase es obtener un prototipo inicial, aquí el desarrollador debe concentrarse en construir un sistema con la máxima funcionalidad, poniendo énfasis en la interfaz del usuario.

2.3.- Evaluación

Los objetivos de esta etapa son obtener por parte de los usuarios la especificación de los requerimientos adicionales del sistema y verificar que el prototipo desarrollado lo haya sido en concordancia con la definición de requerimientos del sistema.

2.4.- Modificación.

Se da cuando la definición de requerimientos del sistema es alterada en la etapa de evaluación.

3.- Diseño técnico.

En esta etapa el sistema debe ser rediseñado y tener la respectiva documentación guiándose en los estándares que tiene la organización.

4.- Programación y prueba

- En esta etapa es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.
- Las pruebas serán de realizarse tantas veces sea necesarias para verificar cualquier tipo de anomalía en el sistema.

5.- Operación

En esta fase se realiza ya la instalación y mantención del software, la complejidad en este caso resulta menor ya que en las etapas anteriores los usuarios han trabajado con el sistema al momento de hacer las pruebas de prototipos

2.1.2 Características, Ventajas y Desventajas de la metodología Prototipado [2].

Características:

- Describe las fases principales de desarrollo de software.
- Define las fases primarias esperadas de ser ejecutadas durante esas fases.
- Ayuda a administrar el progreso del desarrollo del software
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Ventajas:

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina
- Se puede reutilizar el código.

Desventajas:

- El cliente puede quedar convencido con las primeras versiones y, quizás, no vea la necesidad de completar el sistema o rediseñarlo con la calidad necesaria.
- Surgen cambios imprevistos que retrasan el progreso del prototipo.

2.2 Herramientas tecnológicas utilizadas.

Las herramientas a utilizar para este proyecto son:

2.2.1 XAMPP [3]

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor web libre.

- Ventajas
 - ❖ Su instalación es de lo más sencilla, basta descargarlo, extraerlo y comenzar a usarlo. Nos permite instalar el entorno MySQL, Apache y PHP, suficiente para empezar proyectos web o revisar alguna aplicación localmente. Además, trae otros servicios como servidor de correos y servidor FTP
- Desventajas
 - ❖ No soporta MySQL desde la consola.
 - ❖ No se pueden actualizar individualmente las versiones de los programas que instala.
 - ❖ Dificultad para configurar aplicaciones de terceros.
 - ❖ Una desventaja es la garantía.

2.2.2 LAMP [4]

El nombre original LAMP responde a Linux, Apache, MySQL y PHP/Python/Perl, pero también puede incluir otras tecnologías. La idea importante de LAMP es que funciona sobre sistemas operativos Linux y te permite construir apps web que son gestionadas por un servidor web (Apache, Nginx u otros) vía HTTP o HTTPS, en el puerto 80 o 443. Este es el paradigma que determina la arquitectura de la aplicación. En las tecnologías LAMP esto se consigue mediante la unión de las siguientes tecnologías

- Linux, el sistema operativo;
 - Apache, el servidor web;
 - MySQL, el gestor de bases de datos;
 - Perl, PHP, o Python, lenguajes de programación.
- Ventajas
 - ❖ Es una tecnología madura
 - ❖ Está bien documentada
 - ❖ Funciona bien en apps web de tamaño medio y grande donde la gestión de los datos en tiempo real no es vital
 - ❖ Tiene una comunidad muy grande
 - ❖ PHP es el lenguaje de servidor más utilizado del mundo, lo usan WordPress, Drupal, Prestashop, Magento, Moodle y un largo, largo etcétera
 - Desventajas
 - ❖ No funciona bien en aplicaciones de tiempo real que manejan grandes cantidades de datos de forma concurrente
 - ❖ No está de moda
 - ❖ Es más lento que MEAN
 - ❖ Emplea varios lenguajes de programación

2.2.3 PHP [5]

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Podemos saber algo más sobre la programación del servidor y del cliente en el artículo qué es DHTML.

➤ Ventajas

- ❖ Es un lenguaje multiplataforma.
- ❖ Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ❖ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- ❖ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).

➤ Desventajas

- ❖ Como es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código, pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución.
- ❖ El lugar más seguro para ejecutar una aplicación es en un servidor propio, por lo cual, si un cliente o usuario requiere su código en su pc, se tiene que dejar el código, sin manera de ocultarlo, aunque hay muchas aplicaciones que nos ayuda a encriptar el código fuente.
- ❖ Se debe saber al menos HTML para poder hacer un trabajo medianamente funcional.
- ❖ Si no se configura correctamente deja abiertas muchas brechas de seguridad.
- ❖ Se necesita instalar un servidor web.

2.3 Razones por las cuales se eligió esta metodología

La metodología de prototipado permite entregar al usuario lo que sería una visión la solución final en etapas tempranas del desarrollo, reduciendo tempranamente los costos de especificaciones erróneas. Otra de las razones para usar un prototipo es cuando el modelo de fases análisis, diseño es inapropiado, es decir cuando el sistema se lo puede realizar solamente con esta metodología.

3. Resultados

En este capítulo se presenta la implementación de la metodología de prototipado durante el desarrollo del sistema.

3.1 Investigación preliminar

Durante esta etapa se encontró la necesidad de tener una administración de usuarios de sistema que permita tener control sobre el registro de usuarios de red, aplicando como solución formularios de registro

3.2 Especificación de requerimientos y prototipado

En esta fase se realizó una descripción completa del comportamiento del sistema que se planeó desarrollar.

3.2.1 Análisis y especificación

Esta fase consiste en identificar las necesidades que tiene la AGC, donde se obtienen y detallan los objetivos y requisitos iniciales para determinar la viabilidad del sistema como se observa en la tabla 1.

No	Requerimiento
RF1	El sistema deberá permitir el inicio de sesión, por medio de un usuario y contraseña.
RF2	El sistema deberá permitir agregar Administradores de sistema
RF3	El sistema deberá permitir modificar la contraseña de los Administradores de sistema.
RF4	El sistema deberá permitir eliminar Administradores de sistema.
RF6	El sistema deberá mostrar todos los usuarios Administradores
RF7	El sistema deberá permitir modificar los datos de 'Rol' y 'Activo' en los usuarios de red.
RF8	El sistema deberá permitir eliminar usuarios de red
RF9	El sistema deberá permitir agregar usuarios de red.

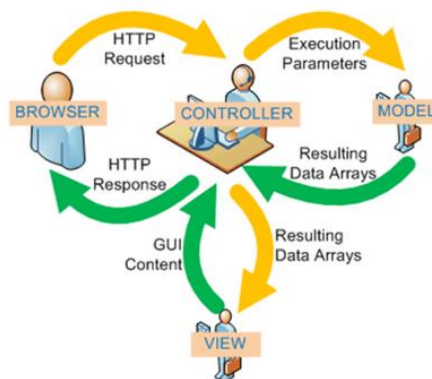
Tabla 1 Requerimientos Funcionales.

3.2.2 Diseño y construcción.

Esta fase del ciclo del desarrollo del prototipo utilizamos la información que se recolectó en la etapa anterior en donde se definieron los procesos, elaboramos el diseño lógico y realizamos la construcción de la aplicación. Además, definimos la arquitectura la cual es MVC (MODELO VISTA CONTROLADOR), es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista** o interfaz de usuario que se compone la información que se muestra al cliente y los mecanismos interacción con éste.
- El **Controlador** que es la parte que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El flujo que sigue el control se muestra en la Figura 3:

**Figura 3.** Flujo que sigue MVC.

En las figuras 4, 5, 6,7 y 8, se muestra el prototipo de baja fidelidad que se desarrolló para aclarar los requerimientos.

Figura 4. Login de Baja Fidelidad.

Matricula	Nombre	Apellido paterno	Apellido materno	Fecha	Rol	Departamento	Activo		
83947563	Andrea	Perez	Lopez	2019-08-09	Alumno		<input type="checkbox"/>	Actualizar	Eliminar
9348463	Juan	Perez	Sanches	2019-08-08	Alumno		<input checked="" type="checkbox"/>	Actualizar	Eliminar
34567	sdfghj	sdfghj	dfghjk	2019-08-13	Practicas		<input checked="" type="checkbox"/>	Actualizar	Eliminar

Figura 5. Menú del sistema de baja fidelidad

Figura 6. Modificar usuario.

Sistema AGC

Bienvenid@, Angel Lopez

Eliminar información de usuario

Matricula	Nombre	Apellido paterno	Apellido materno	Fecha registro	Rol	Departamento	Activo
83947563	Andrea	Perez	Lopez	2019-08-09	Alumno		NO

[Regresar](#)

Figura 7. Eliminar usuario

Registro de usuario de red

Matricula

Nombre:

Apellido paterno:

Apellido materno:

Rol:

[Regresar](#)

Figura 8. Registro de nuevo usuario de red.

3.2.3 Evaluación

Las necesidades o requerimientos identificados en la AGC, se cubrieron satisfactoriamente con el sistema, se puede apreciar en la figura 9 existe la parte de Registrar, Eliminar, Actualizar, Buscar, la diferenciación de activo e inactivo.

Panel de administración [Registrar](#) [Estadísticas](#) [Cerrar sesión](#)

Bienvenid@, Angel Lopez

Sistema AGC

Buscar:

Usuarios de Red

Matrícula	Nombre	Apellido paterno	Apellido materno	Fecha	Rol	Departamento	Activo			
83947563	Andrea	Perez	Lopez	2019-08-09	Alumno		<input type="checkbox"/>	PDF	Actualizar	Eliminar
9348463	Juan	Perez	Sanches	2019-08-08	Alumno		<input checked="" type="checkbox"/>	PDF	Actualizar	Eliminar
34567	Luz	Perez	Juarez	2019-08-13	Alumno		<input checked="" type="checkbox"/>	PDF	Actualizar	Eliminar

Figura 9. Requisitos evaluados.

3.2.4 Modificación.

En esta fase se modificó el prototipo de acuerdo a los comentarios que se realizaron por parte de la AGC. En la figura 11 se aprecia la modificación de la interfaz del Login.

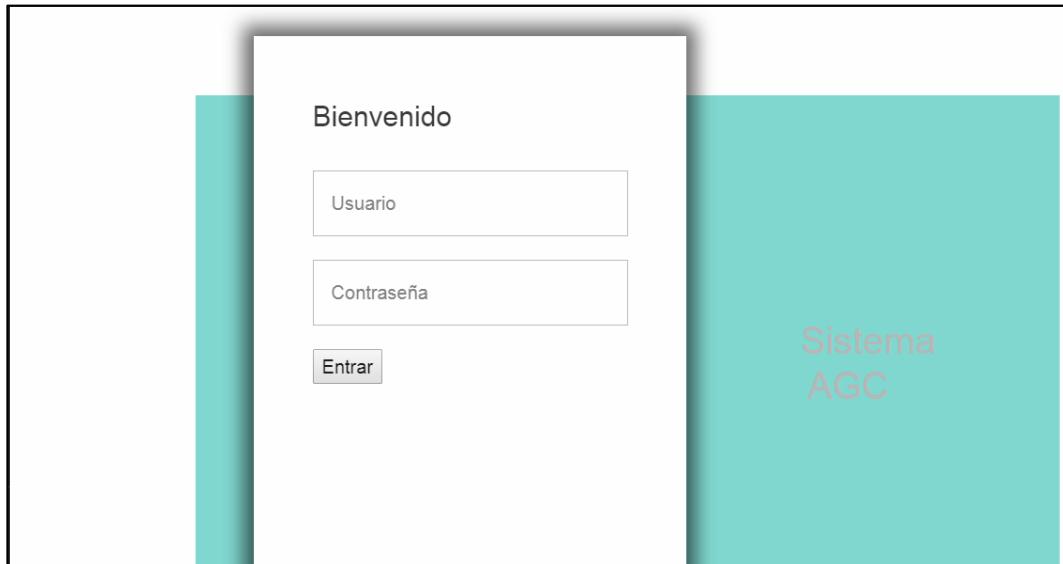


Figura 11. Interfaz de Login.

3.3 Diseño Técnico

En esta etapa se mostrarán los casos de uso del sistema, figuras 12, 13, 14.

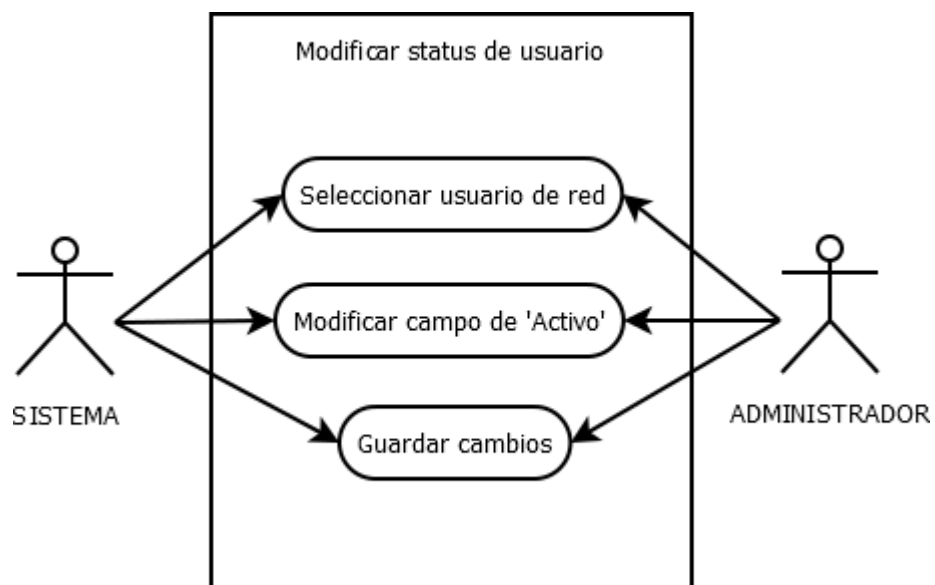


Figura 12. Caso de uso Modificar.

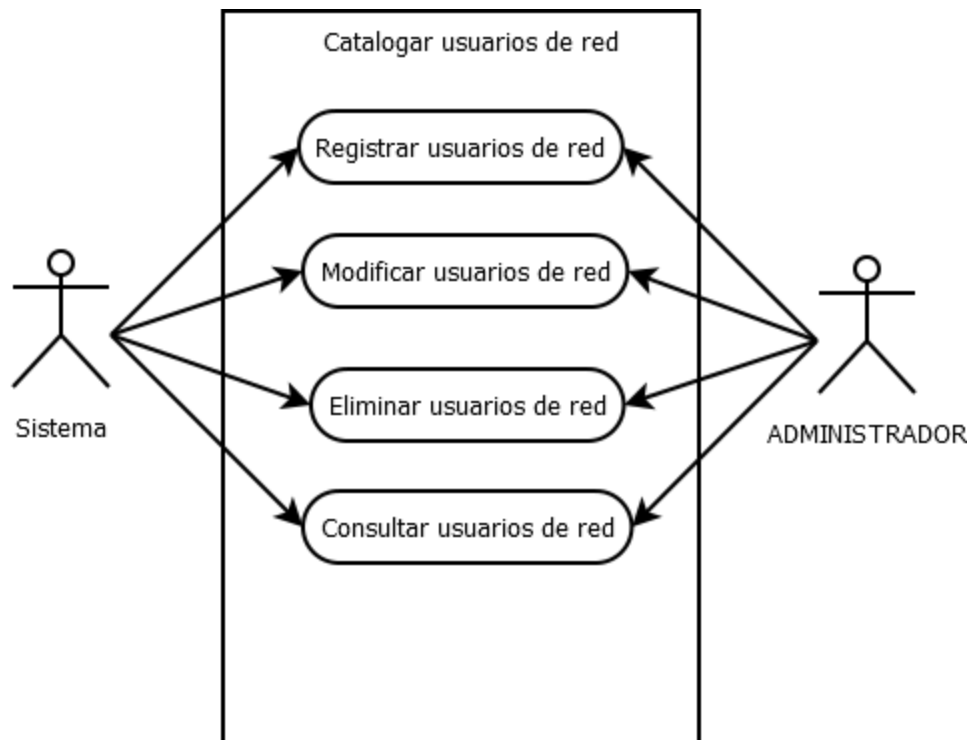


Figura 13. Caso de uso de usuarios de red.

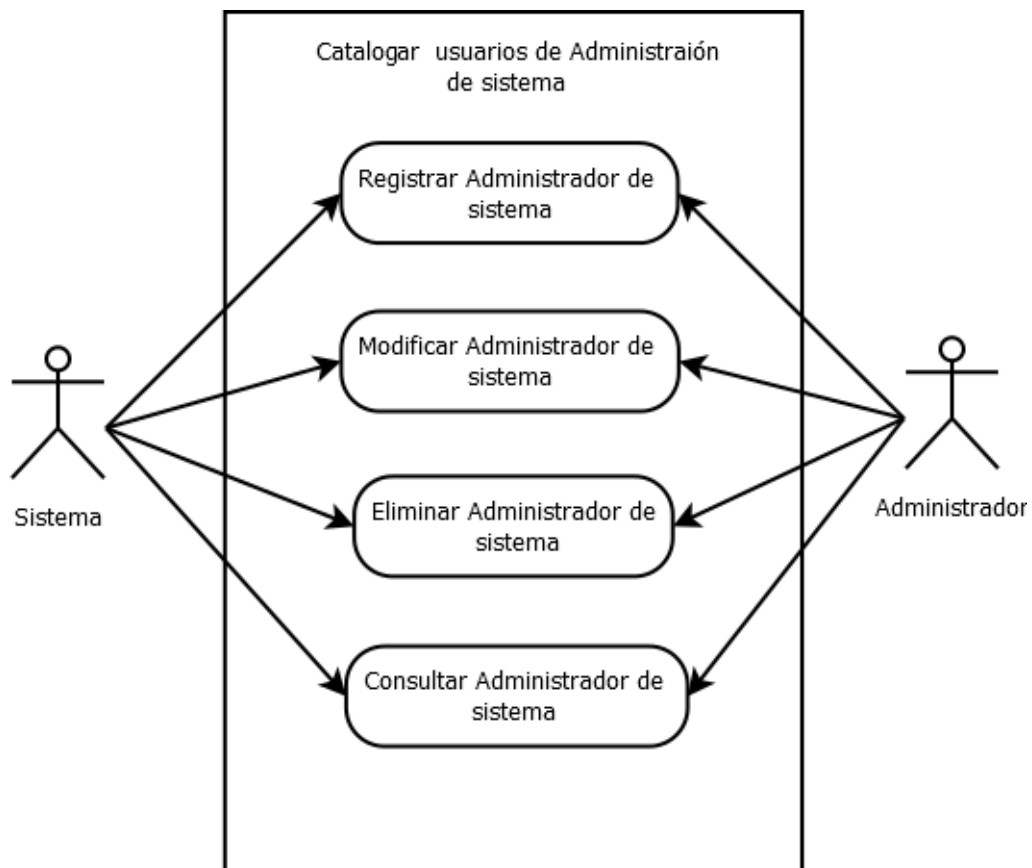


Figura 14. Caso de uso de usuarios Administradores de sistema

En las figuras 15, 16, 17, 18,19,20,21,22 y 23 se muestran los diagramas de actividad.

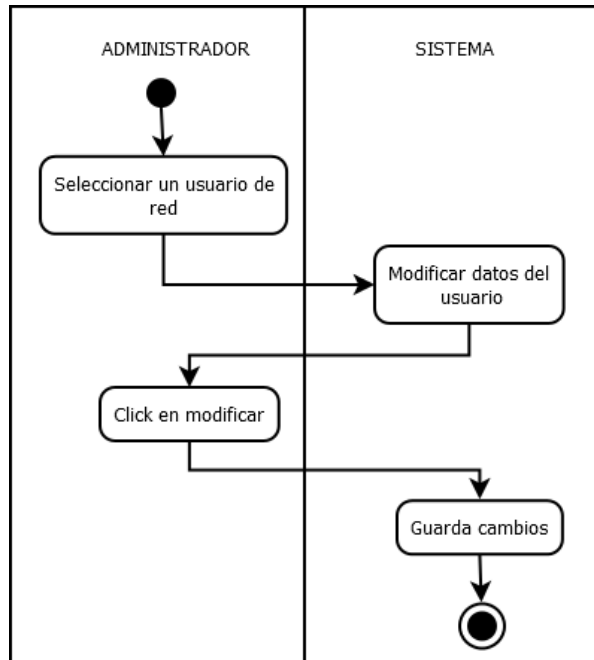


Figura 15. Diagrama de actividad de la interfaz de actualizar.

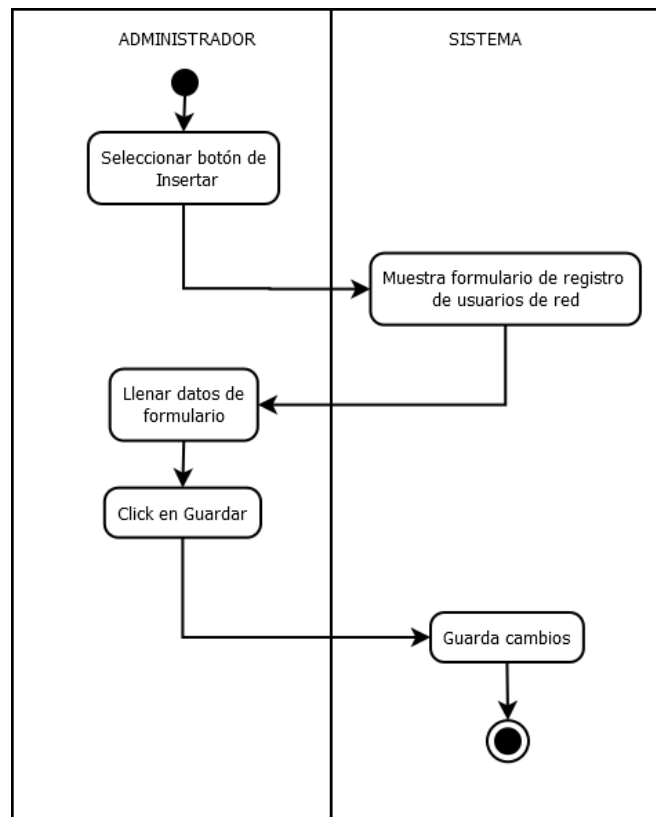


Figura 16. Diagrama de actividad de la interfaz de insertar usuarios de red.

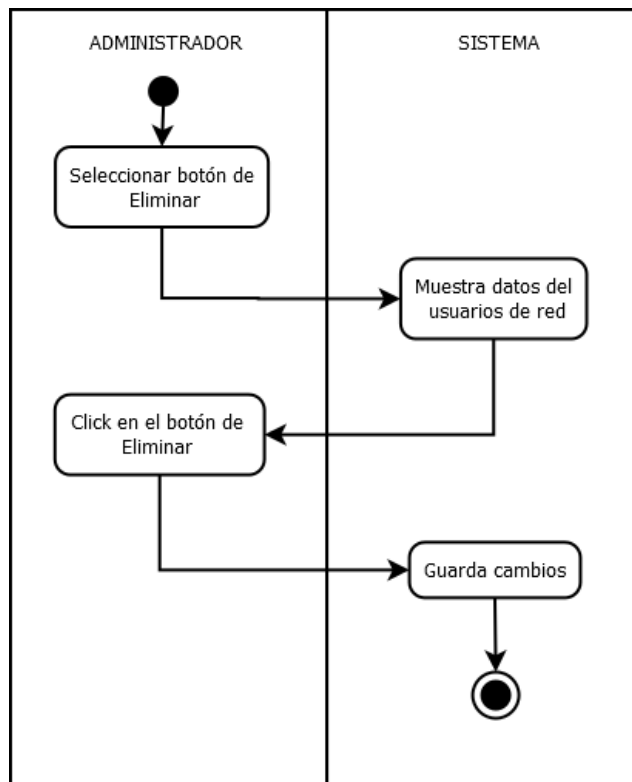


Figura 17. Diagrama de actividad de la interfaz de eliminar usuario de red. solicitado

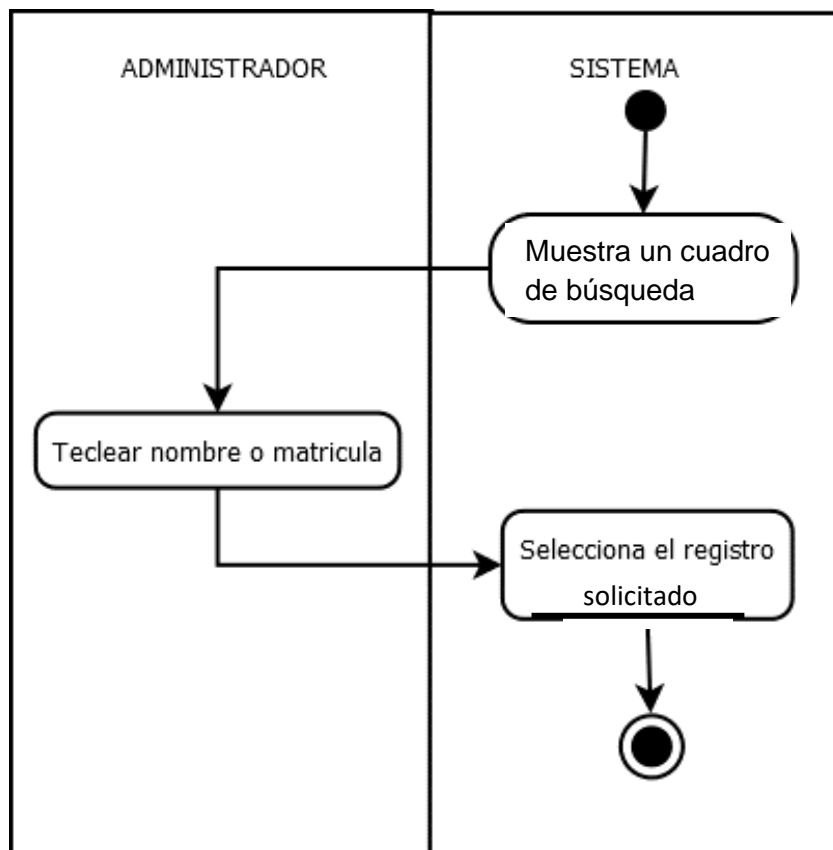


Figura 18. Diagrama de actividad de la interfaz de búsqueda de usuario de red.

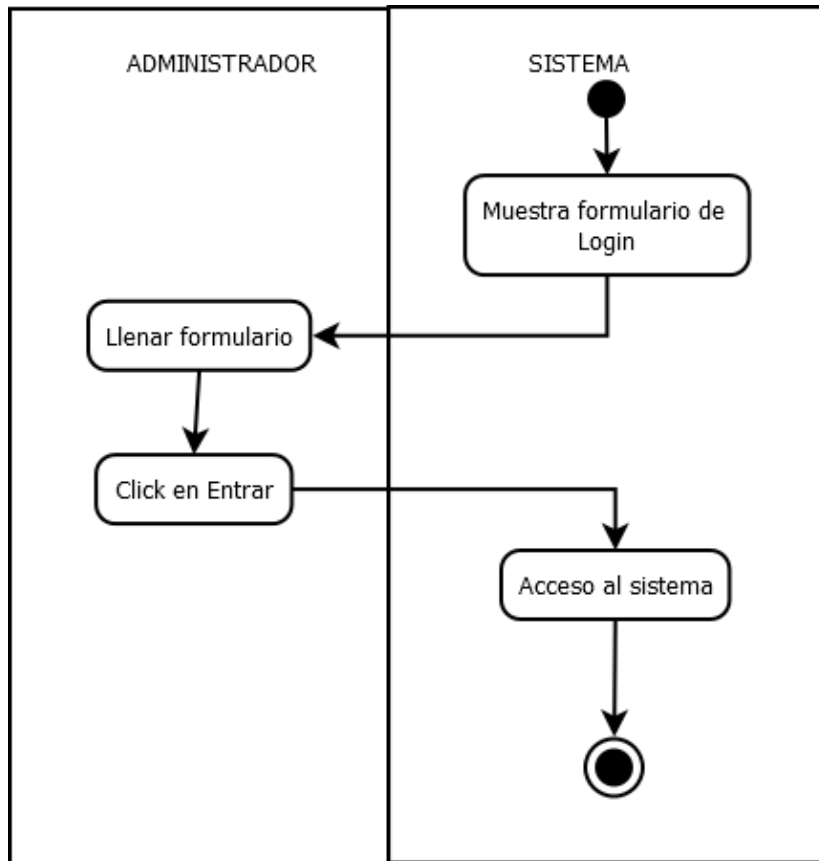


Figura 19. Diagrama de actividad de la interfaz de login.

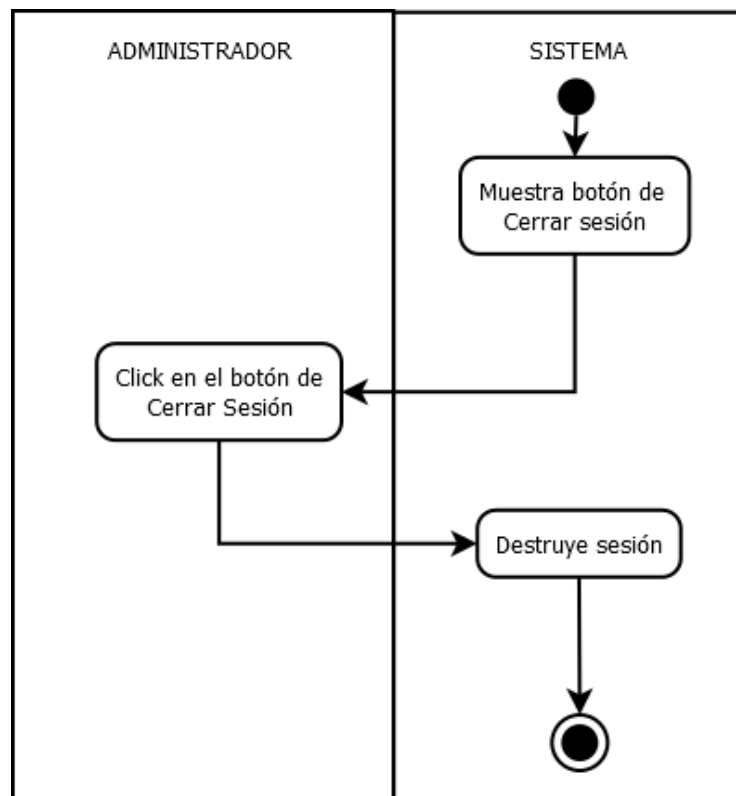


Figura 20. Diagrama de actividad de Cerrar sesión.

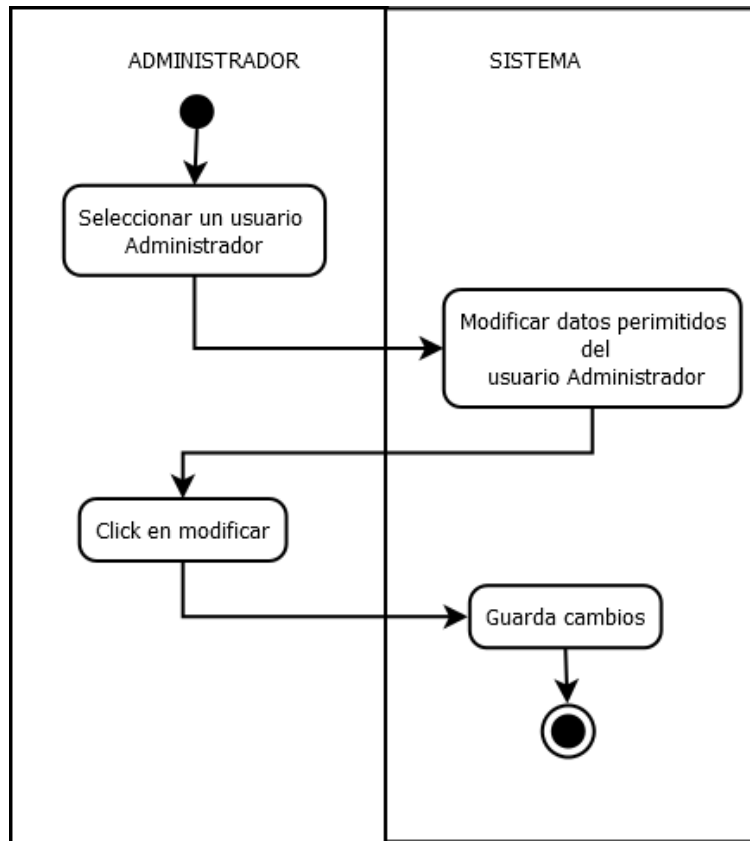


Figura 21. Diagrama de actividad de la interfaz de modificar un administrador.

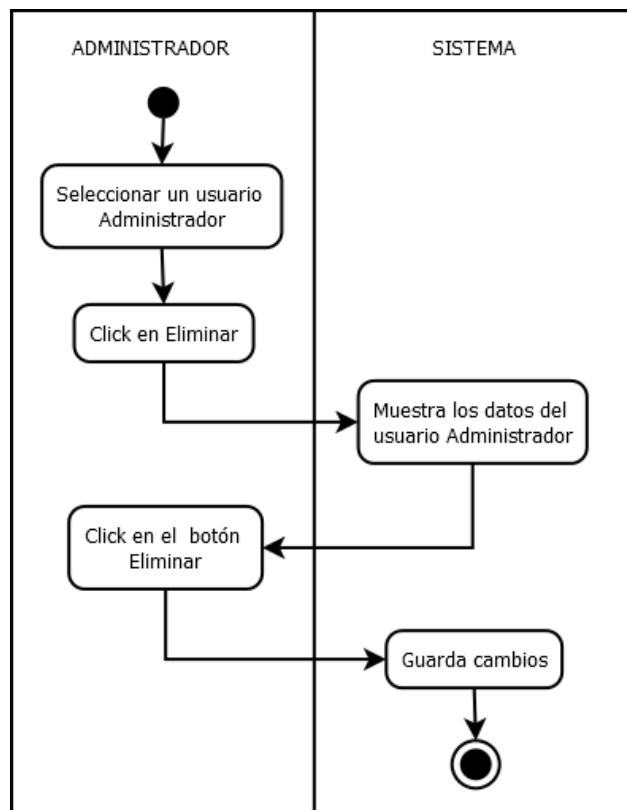


Figura 22. Diagrama de actividad de la interfaz de eliminar administrador.

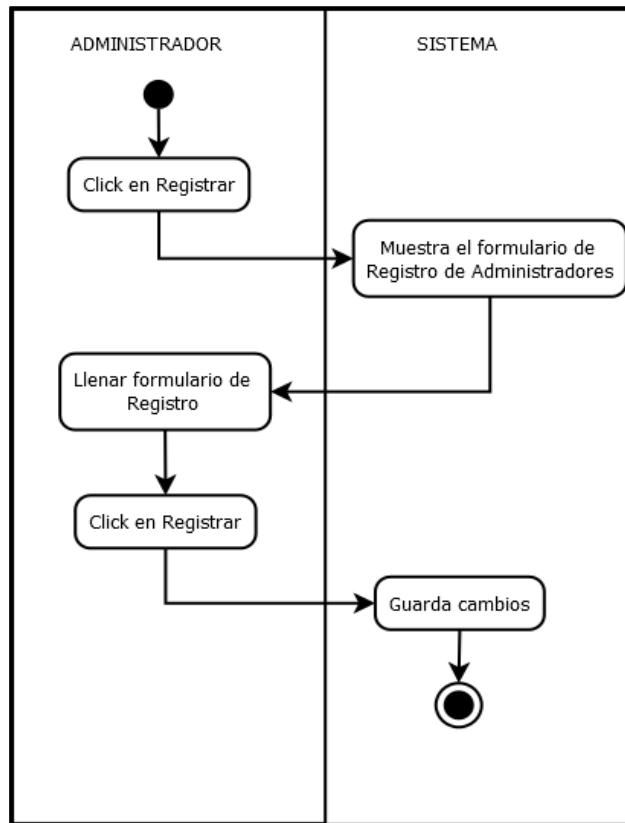


Figura 23. Diagrama de actividad de la interfaz de registro de administradores.

3.4 Programación y Prueba

En la Figura 24 se muestra el diseño de E-R de la base de datos, sobre el cual se trabaja el sistema.

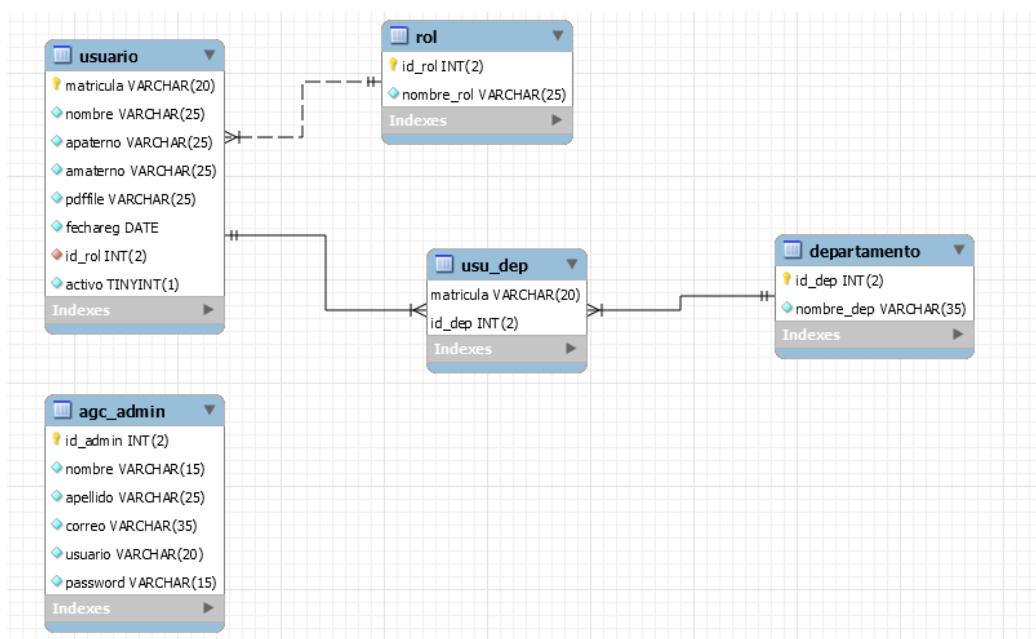


Figura 24. Diagrama Entidad Relación del Proyecto.

La Figura 25 muestra de la interfaz de todos los usuarios de red, contiene la parte de activo, eliminación, modificación, actualización y el registro de los usuarios de red.



Figura 25. Interfaz de Menú.

La Figura 26 muestra la interfaz que realiza el registro de un usuario de red.

Sistema AGC

Registro de usuario de red

Matricula

Nombre:

Apellido paterno:

Apellido materno:

Rol:

Figura26. Interfaz de Registro de usuarios de red.

La figura 27 muestra la interfaz de la eliminación de un usuario de red.

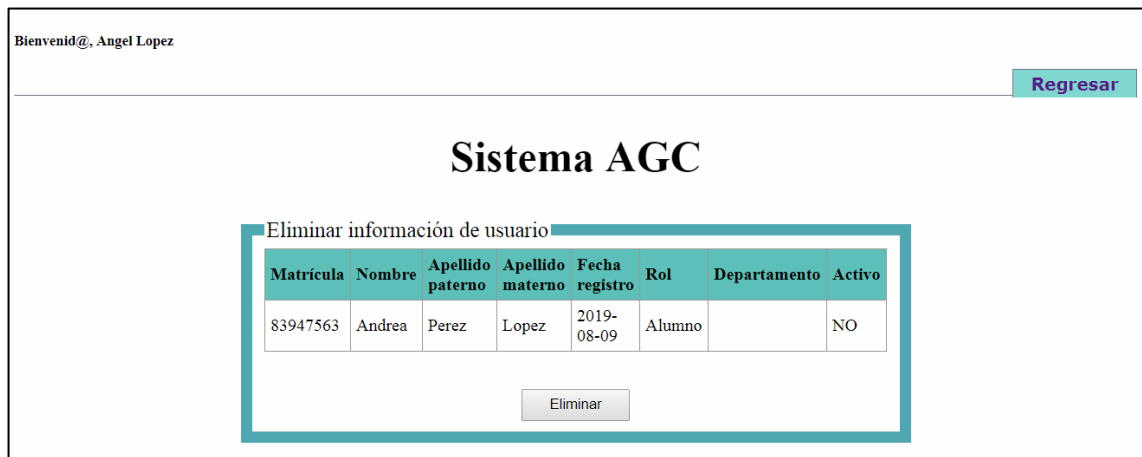


Figura 27. Interfaz de eliminar usuarios de red.

La Figura 28 muestra la interfaz que modifica el 'Rol' y 'Activo' del usuario de red.

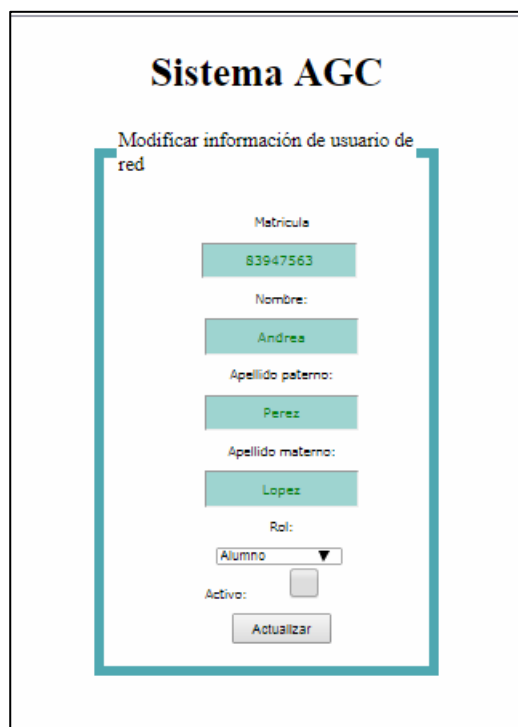


Figura 28. Interfaz de actualizar usuarios de red.

Las pruebas se realizaron de manera exitosa, bajo la supervisión del asesor técnico como se muestra en la Figura 29, considerando los aspectos de fallas en el uso del sistema en una prueba piloto.

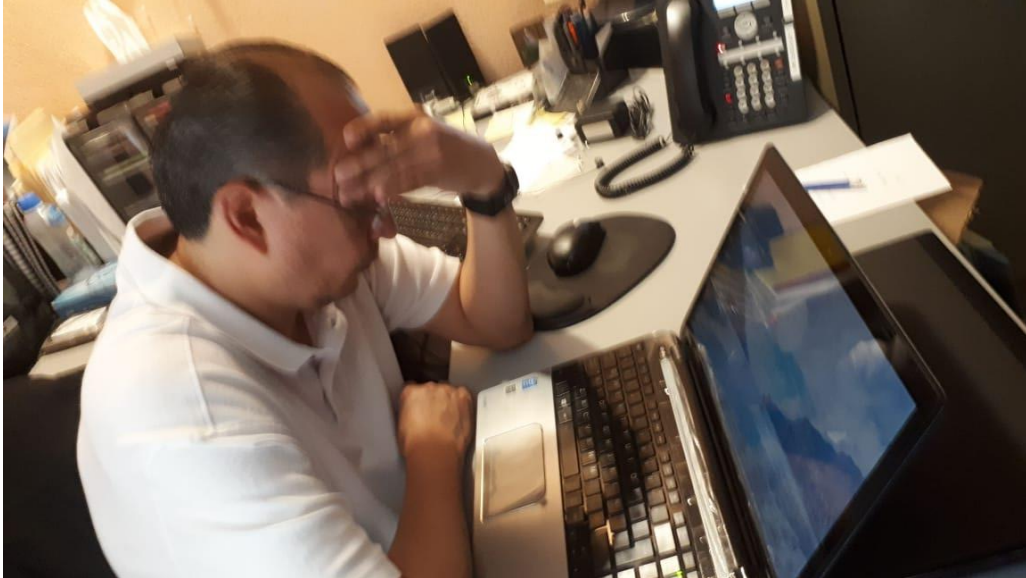


Figura 29. Asesor técnico usando el sistema en prueba piloto

3.5 Operación

En esta fase se ha configurado el sistema para mejorar la administración en caso de requerir implementar más procesos, tomando en cuenta que no hubo peticiones de correcciones al haberse demostrado su rendimiento y calidad del funcionamiento del sistema

4. Conclusiones y recomendaciones

Este proyecto realizado en el INAOE me sirvió para conocer y generar una idea más clara sobre el desarrollo de sistemas a nivel empresarial, dado que este tipo de conocimiento fue generado en clases de ingeniería, con el que actualmente aprendí a aprovechar durante el desarrollo de este sistema.

Al principio existieron muchas dificultades para realizar el proyecto, tales como: la escasa experiencia del lenguaje php utilizado en POO para poder realizar el sistema completo, además de la inexperiencia con el uso del patrón de diseño MVC, en donde las primeras versiones del sistema fueron incorrectas al desarrollar la interacción del modelo, vista y controlador. Este proyecto también consistió en mantener mucha comunicación con el asesor, ya que se le mantenía al tanto de los cambios realizados durante todo el tiempo de desarrollo.

Durante el desarrollo del sistema encontré distintas formas de satisfacer la interacción del usuario, al principio se contemplaban distintas interfaces para cada requerimiento, pero conforme se presentaban los resultados iba notando que era factible realizar mas de 2 actividades por cada interfaz.

El sistema lo puede continuar trabajando cualquier persona que desee agregarle funcionalidades relacionadas a las que hace actualmente, lo único que se les recomienda es seguir con el correcto nombramiento de cada archivo tal cual lo usa el sistema y usar de forma correcta el patrón de diseño MVC.

Finalmente, este proyecto me fue útil para conocer y poner en práctica la POO con php, además de mejorar mi programación con nuevos métodos.

5. Referencias bibliográficas

[1] Breton, J. García, G & Rojas, I. "Reseña del Modelo de Prototipo y Herramientas Case". Mayo 2011. Disponible en: <http://gestionrrhusm.blogspot.com/2011/05/modelo-de-prototipo.html>

[2] URL: <https://ingsoftware.weebly.com/ciclo-de-vida-de-un-prototipo.html> Ciclo de vida de un Prototipo. Fecha de consulta 28/06/2019.

[3] URL: <https://www.apachefriends.org/es/index.html> Página principal de Apache Friends, en ella se puede consultar la descripción del entorno de desarrollo PHP y distribución de XAMPP, realizado por Kai 'Oswald' Seidler, Kay Vogelgesang. Fecha de consulta: 01/07/2019

[4] URL: <https://bitnami.com/stack/lamp> página principal de Bitnami, en ella se puede consultar la información sobre LAMP y sus características. Fecha de consulta: 01/07/2019

[5] URL: <https://www.php.net/manual/es/intro-what-is.php> página principal de PHP en ella se puede consultar la información sobre este lenguaje. Fecha de consulta: 01/07/2019



Universidad Politécnica de Puebla
Ingeniería en Informática

Lourdes Nataly Rojas Hernández
Víctor Manuel Cortés Castillo
Rebeca Rodríguez Huesca

Este documento se distribuye para los términos de la
Licencia 2.5 Creative Commons (CC-BC-NC-ND 2.5 MX)