



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

**SISTEMA INTEGRAL DE ADMINISTRACIÓN Y CONTROL
DE USUARIOS ONLINE**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTA:

JUÁREZ GARCÍA IOVANI

DIRECTOR:

ING. ROMÁN OSORIO COMPARÁN



Ciudad Universitaria, México, D.F., Marzo 2013.

Dedicatorias

*A mi madre,
Rosa María García Alarcón,
Por tu apoyo sin medida ni condición, por darme la vida y enseñarme a
apreciarla.*

*A mi padre,
Rubén Juárez González,
Por tu ejemplo y dedicación que motivaron este trabajo.*

*A mi hermano,
Miguel Juárez García,
Por compartir el camino y como ejemplo para que continúes el tuyo.*

Agradecimientos

*A mis familiares y amigos,
Por sus consejos y su apoyo durante toda la carrera.*

*A mis profesores y compañeros,
Por las lecciones aprendidas dentro y fuera del salón de clases.*

*Al Ingeniero Román Osorio Comparán,
Por la oportunidad y el honor de contar con su asesoría.*

*A los miembros del jurado,
Por su tiempo e interés indispensables para terminar este trabajo.*

ÍNDICE

✚	Objetivo	2
✚	Introducción	3
✚	Capítulo 1. Marco teórico	4
▣	1.1 Conceptos Teóricos	5
▪	Sistemas Online	5
▪	Bases de datos	5
○	Tipos de bases de datos	5
○	Modelos de bases de datos	6
▪	Sistemas de Administración de Bases de Datos	7
▪	Seguridad Informática	8
▪	Administración de Usuarios	9
▪	Criptografía	9
○	Sistemas de cifrado modernos	9
○	AES	10
➤	Pseudocódigo	10
➤	Descripción	12
▪	Ingeniería de Software	13
○	Marco de trabajo para el proceso	13
▪	Lenguajes de programación	15
○	Paradigmas	15
○	Programación orientada a objetos	16
➤	Características de la POO	17
▪	Hojas de estilo en cascada	18
○	Sintaxis	18
○	Ventajas	19
▣	1.2 Información	19
▣	1.3 Metodología y estrategia de desarrollo	19
▪	Modelo en cascada	20
▣	1.4 Herramientas de desarrollo	21
▪	JDeveloper	21
▪	MySQL	21
▪	Bizagi Process Modeler	21
▪	Dia	22
▣	1.5 Estado del arte. Alternativas de solución	22
▪	Single Sign-On	22
✚	Capítulo 2. Diseño del Sistema	24
▣	2.1 Requerimientos	25
▣	2.2 Alcance	25
▣	2.3 Proceso	25
▣	2.4 Casos de uso	27
▪	Diagrama	27
▪	Realización de casos de uso	28
○	Registra solicitud	28
○	Login	29
○	Autoriza cuenta de usuario	30
○	Cancela cuenta de usuario	31

○	Cambia datos de usuario	32
○	Cambia contraseña	33
○	Solicita aplicación y rol	34
○	Solicita baja de cuenta de usuario	35
○	Accede a aplicación	36
❏	2.5 Diagrama entidad-relación	37
❏	2.6 Diccionario de datos	38
▪	Tabla USERS	39
▪	Tabla SISTEMAS	41
▪	Tabla TIPOS_ROLES	42
▪	Tabla CONTROL_ACCESOS	43
▪	Tabla OPCIONES_SISTEMA	44
▪	Tabla MENUS	45
▪	Tabla OPCIONES_MENU	46
▪	Tabla FACULTADES	47
✚	Capítulo 3. Desarrollo del sistema	48
❏	3.1 Autenticación de usuarios	52
❏	3.2 Generación de responsiva PDF	54
❏	3.3 Registro de usuarios	58
❏	3.4 Recuperación de login	59
❏	3.5 Menú de usuarios	63
▪	Administración de usuarios	63
▪	Inicio	64
▪	Sistemas	65
▪	Servicios	66
▪	Salir	67
✚	Capítulo 4. Implementación, pruebas y resultados	68
❏	4.1 Base de datos	69
❏	4.2 Servidor de aplicaciones	73
❏	4.3 Pruebas	73
▪	Registro de usuarios	74
▪	Inicio de sesión	76
▪	Actualización de datos de usuario	77
▪	Solicitud de sistema y rol	77
▪	Cambio de contraseña	78
❏	4.4 Resultados	80
✚	Capítulo 5. Conclusiones	81
✚	Bibliografía y referencias	83

ÍNDICE DE FIGURAS

Figura 1.1 Fase de SubBytes del cifrado AES	10
Figura 1.2 Fase de ShiftRows del cifrado AES	11
Figura 1.3 Fase de MixColumns del cifrado AES	11
Figura 1.4 Fase de AddRoundKey del cifrado AES	11
Figura 1.5 Representación del modelo en cascada	20
Figura 2.1 Flujo de procesos del sistema	26
Figura 2.2 Diagrama de casos de uso	27
Figura 2.3 Diagrama E-R de la base de datos	37
Figura 3.1 Clases, librerías y paginas del sistema	49
Figura 3.2 Clase ManejaBD.java	50
Figura 3.3 Clase ConsultaBD.java	51
Figura 3.4 Pantalla inicial de SIAC-Online	52
Figura 3.5 Código de login.jsp	53
Figura 3.6 Clase ResponsivaPDF.java	54
Figura 3.7 Responsiva en formato PDF	57
Figura 3.8 Página SolicitaUsuario.jsp	58
Figura 3.9 Registro de un nuevo usuario	59
Figura 3.10 RecuperaLogin.jsp	60
Figura 3.11 Correo electrónico	60
Figura 3.12 Clase SMTPAuthentication.java	61
Figura 3.13 Clase EnviaMail.java	61
Figura 3.14 Menú de administrador	63
Figura 3.15 Menú de usuario	63
Figura 3.16 Usuarios registrados	63
Figura 3.17 Administración de sistemas del usuario	64
Figura 3.18 información del usuario	64
Figura 3.19 Formulario de actualización de datos	65
Figura 3.20 Sistemas habilitados	65
Figura 3.21 Solicitud de rol	66
Figura 3.22 Formulario de cambio de contraseña	66
Figura 3.23 Responsivas	67
Figura 3.24 Cierre de sesión	67
Figura 4.1 Código de creación de tablas	69
Figura 4.2 Código de inserción de datos	71
Figura 4.3 Configuración de servidor de aplicaciones	73
Figura 4.4 Campos obligatorios	74
Figura 4.5 Validación de campos	75
Figura 4.6 Validación de captcha	75
Figura 4.7 Mensaje de error	76
Figura 4.8 Bloqueo de sesión	76
Figura 4.9 Validación de datos	77
Figura 4.10 Solicitud de rol	77
Figura 4.11 Validación de contraseña	78
Figura 4.12 Validación de contraseña sin caracteres especiales	78
Figura 4.13 Validación de contraseña anterior	79
Figura 4.14 Validación de contraseña nueva	79

ÍNDICE DE TABLAS

✚ Tabla 1.1 Implementaciones de Single Sign-On	23
✚ Tabla 2.1 Relación de tablas de base de datos	38
✚ Tabla 2.2 Detalle de tabla USERS	39
✚ Tabla 2.3 Llaves de tabla USERS	40
✚ Tabla 2.4 Detalle de tabla SISTEMAS	41
✚ Tabla 2.5 Llaves de tabla SISTEMAS	41
✚ Tabla 2.6 Detalle de tabla TIPOS_ROLES	42
✚ Tabla 2.7 Llaves de tabla TIPOS_ROLES	42
✚ Tabla 2.8 Detalle de tabla CONTROL_ACCESOS	43
✚ Tabla 2.9 Llaves de tabla CONTROL_ACCESOS	43
✚ Tabla 2.10 Detalle de tabla OPCIONES_SISTEMA	44
✚ Tabla 2.11 Llaves de tabla OPCIONES_SISTEMA	44
✚ Tabla 2.12 Detalle de tabla MENUS	45
✚ Tabla 2.13 Llaves de tabla MENUS	45
✚ Tabla 2.14 Detalle de tabla OPCIONES_MENU	46
✚ Tabla 2.15 Llaves de tabla OPCIONES_MENU	46
✚ Tabla 2.16 Detalle de tabla FACULTADES	47
✚ Tabla 2.17 Llaves de tabla FACULTADES	47

**Sistema Integral de Administración y
Control de Usuarios
Online**

SIAC – Online

Objetivo

Unificar el acceso de los usuarios a las diferentes aplicaciones asignadas tomando en cuenta el rol que tengan dentro de éstas, es decir, de acuerdo a los privilegios de cada usuario.

Registrar nuevos usuarios y modificar o actualizar los ya registrados.

Mantener la base de datos actualizada y mostrar siempre información real.

Aumentar la confiabilidad del usuario y facilitar su administración.

Introducción

El desarrollo de aplicaciones y la seguridad de los mismos, son temas de suma importancia cuando se implementan en instituciones gubernamentales.

El acceso a los servicios que una institución ofrece debe ser regulado y administrado de forma eficiente para garantizar la integridad, disponibilidad y confidencialidad de la información que en esta se maneja. Es por esto que la creación de usuarios y roles dentro de cada sistema es necesaria para evitar fuga de información y exponer información sensible a cualquier entidad.

Sin embargo, el desarrollo de nuevas aplicaciones conlleva la creación de usuarios para cada una de éstas lo que a su vez implica el consumo de espacio tanto físico como lógico de bases de datos, infraestructura de red, almacenamiento, etc., haciendo tediosa la tarea de ingresar nombre de usuario y contraseña para cada aplicación o sistema que sea solicitado.

El Sistema Integral de Administración y Control de Usuarios Online centraliza el acceso a través de un único login a partir del cual será posible abrir o solicitar las diferentes aplicaciones o sistemas que estén disponibles para el usuario de acuerdo con el rol que tenga asignado.

Con esto, la administración de usuarios y el control de sus accesos se facilitan puesto que toda su información será almacenada en una base de datos única.

Actualmente, es posible implementar en cada aplicación un esquema propio de autenticación y gestión de usuarios, sin embargo esto implica que para cada uno se debe escribir código nuevo y usar espacio adicional en la base de datos para almacenar la información de usuarios.

A nivel institucional, esto puede significar un problema puesto que no siempre se tiene el control del espacio en disco o las licencias adecuadas para la gestión de estos recursos.

CAPÍTULO 1
MARCO TEÓRICO

1.1 Conceptos teóricos

Sistemas online

Online (del inglés) significa “en línea”. El concepto se refiere, en el ámbito de la informática, a un sistema, aplicación o información que es visible a través de una red, generalmente Internet. Algunos sistemas online permiten acceder a bases de datos e interactuar con estas.

Algunas ventajas de trabajar con sistemas online son las siguientes:

- Información más rápida y precisa
- Control de costos
- Seguridad
- Portabilidad
- Integración
- Interfaz amigable

Bases de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Existen programas denominados Sistemas Gestores de Bases de Datos, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Tipos de base de datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

Según la variabilidad de los datos almacenados

- Bases de datos estáticas
- Bases de datos dinámicas

Según el contenido

- Bases de datos bibliográficas
- Bases de datos de texto completo
- Directorios
- Bases de datos o "bibliotecas" de información química o biológica

Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

- Bases de datos jerárquicas
- Base de datos de red
- Bases de datos transaccionales
- Bases de datos relacionales
- Bases de datos multidimensionales
- Bases de datos orientadas a objetos
- Bases de datos documentales
- Bases de datos deductivas

Sistemas de administración de bases de datos

Un Sistema de Administración de Bases de Datos (DBMS) es un software que define una base de datos, guarda los datos, permite un lenguaje de consulta, genera informes y crea pantallas para ingresar datos.

Componentes de un sistema de administración de base de datos

- **Motor de la base de datos:** es el elemento principal que interviene en el desempeño y la capacidad del sistema. Su responsabilidad es guardar, recuperar y actualizar los datos.
- **Diccionario de datos:** contiene las definiciones de todas las tablas de datos. Describe los tipos de datos que se guardan, permite al DBMS dar seguimiento a los datos.
- **Generador de formularios:** sirve de ayuda para el desarrollador a crear los formularios de ingreso que tienen como finalidad representar las tareas comunes de un usuario.
- **Generador de aplicaciones:** consta de herramientas que ayudan al desarrollador a crear una aplicación completa, tales como generador de menús y barras de herramientas así como un sistema de ayuda contextual integrado.
- **Comunicación e integración:** es la red de comunicaciones que conecta la base de datos en diferentes equipos incluso en diferentes lugares, además, el conector de 3GL permite acceder a la base a través de lenguajes de programación tradicionales
- **Seguridad:** debido a que la base de datos comparte información entre los diferentes usuarios, es necesario que el DBMS también administre la seguridad mediante controles de acceso.

La administración de datos consiste en la planificación y coordinación requerida para definir datos de manera congruente en toda la compañía. Una persona o grupo debe tener la responsabilidad de determinar los datos que deben reunirse, como deben guardarse y fomentar modos de utilizar dichos datos. Esta persona o grupo es responsable por la integridad de los datos.

La administración de una base de datos consta de los aspectos técnicos de creación y ejecución de la base de datos. Las tareas básicas son el monitoreo del desempeño, el respaldo y la recuperación, así como la asignación y control de la seguridad. Los administradores de bases de datos están capacitados en los detalles de instalación, configuración y operación del DBMS.

Seguridad informática

La interconexión de computadoras personales y estaciones de trabajo con grandes sistemas y aplicaciones de grandes bases de datos, plantea un nuevo problema de seguridad al repartir los recursos y no tener una gestión centralizada.

Por otra parte, las repercusiones de un problema en una computadora central a la que estén conectados un gran número de usuarios serán mayores que en un sistema distribuido, donde un fallo grave en un equipo provocaría que solo una parte de la red quedase afectada.

La cantidad de información que circula en una red informática es enorme y de valor incalculable, de ahí la importancia de su seguridad. Cuando se define la seguridad adecuada para cada caso, se debe hacer un estudio de los riesgos, tomando en cuenta factores como el valor de información contenida, las consecuencias de que la red no esté disponible por un determinado periodo de tiempo, los equipos, los sistemas operativos, aplicaciones y gastos de administración necesarios para conseguir el nivel de seguridad deseado.

Los mecanismos más eficaces para combatir las principales amenazas a las que se enfrenta una red como el robo, la distribución o alteración de la información y el acceso no autorizado, son el control de accesos y el cifrado.

Además existen otros métodos frente a distintas amenazas como la realización de copias de seguridad para evitar la destrucción de información ya sea accidental o intencionada, restringir la lectura y/o escritura de ficheros en las estaciones de trabajo para mantener los virus informáticos fuera de nuestros sistemas, etc. En un sistema multiusuario cobra mayor importancia la adecuada formación de los usuarios para mantener la seguridad del sistema.

Otra parte importante en la seguridad de la red es la seguridad física pues es conveniente planear previamente la localización de todos los elementos para que estén en lugares controlados y fuera del alcance de cualquier persona.

Para facilitar la comunicación entre las redes de distintos fabricantes, la Organización Internacional de Estandarización publicó en 1983 la normativa ISO 7498 en donde definió un modelo estándar de Interconexión de Sistemas Abiertos (OSI en inglés) que permite la comunicación entre diferentes sistemas con diferentes equipos y sistemas operativos. En 1988 se publicó la segunda parte de la normativa OSI (ISO 7498-2) en el que se definen los servicios de seguridad de la arquitectura, como la confidencialidad, la integridad y el control de accesos y se señala el cifrado de la información como el mecanismo más adecuado para conseguir los objetivos de seguridad que se pretenden alcanzar.

Administración de usuarios.

En los sistemas o aplicaciones existe un encargado de la gestión de este o administrador de sistema cuyas funciones son muy amplias entre las que se encuentran: gestión de cuentas y claves de usuario, asignación de roles, asignación de permisos y derechos de acceso, respaldo de la base de datos, actualización de datos del sistema y de usuarios, aprovechamiento óptimo de los recursos, implantación de medidas de seguridad.

Para que un usuario identificable tenga acceso al sistema, debe introducir un nombre identificativo y una palabra clave. Una vez que el usuario está dentro del sistema, podrá utilizar sus recursos en función de los permisos y derechos de acceso que el administrador le haya asignado. Los roles simplifican notablemente la tarea de administración y definición de permisos y derechos de acceso para los usuarios.

Criptografía

El término de criptografía proviene de las palabras griegas Kriptos y Graphos que significan secreto u oculto y escribir, respectivamente. Es decir, significa "escritura oculta". Como concepto, se refiere a las técnicas matemáticas utilizadas para cifrar y descifrar información que hagan posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos.

El sistema para escribir un mensaje secreto se denomina criptosistema y el mensaje secreto es el criptograma. A la acción de convertir un mensaje en claro en uno secreto se conoce como cifrado y a la acción inversa se le denomina descifrado.

Sistemas de cifrado modernos

Se clasifican en sistemas de clave privada también conocidos como simétricos o convencionales, y sistemas de clave pública llamados también asimétricos.

En los sistemas de clave privada el emisor o remitente de un mensaje cifra la información con una determinada clave que el receptor o destinatario deberá utilizar para proceder al descifrado. En los sistemas de clave pública se utilizan dos claves, una pública y otra privada. El emisor cifra la información con la clave pública y descifra con su propia clave privada, solo conocida por él y que debe permanecer en secreto.

El remitente del mensaje extrae la clave pública del destinatario accediendo a una base de datos con las claves públicas de los posibles destinatarios. Utilizando esta clave, el mensaje

solo podrá ser descifrado si se posee la clave privada correspondiente; es decir, la que tiene el destinatario.

Estándar de Cifrado Avanzado (AES)

Es un esquema de cifrado por bloques adoptado por el gobierno de los Estados Unidos. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica.

Originalmente llamado "Rijndael", "Rain Doll" en inglés, fue desarrollado por dos criptólogos belgas, Joan Daemen y Vincent Rijmen, ambos estudiantes de la Katholieke Universiteit Leuven.

Estrictamente hablando, AES es una particularidad de Rijndael (aunque en la práctica se los llama de manera indistinta) ya que Rijndael puede ser especificado por una clave que sea múltiplo de 32 bits, con un mínimo de 128 bits y un máximo de 256 bits, mientras que AES tiene un tamaño de bloque fijo de 128 bits y tamaños de llave de 128, 192 o 256 bits.

AES opera en una matriz de 4×4 bytes, llamada estado.

Pseudocódigo

- Expansión de la clave usando el esquema de claves de Rijndael.
- Etapa inicial:
 1. AddRoundKey
- Rondas:
 1. SubBytes — en este paso se realiza una sustitución no lineal donde cada byte es reemplazado con otro de acuerdo a una tabla de búsqueda.

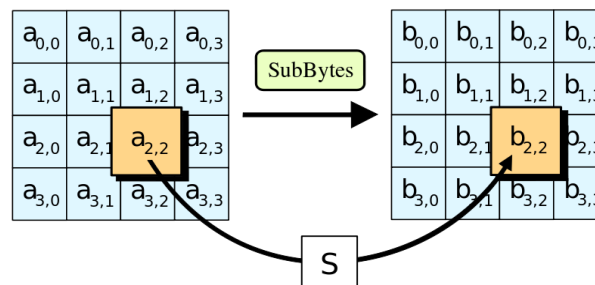


Figura 1.1. Fase de SubBytes del cifrado AES.

2. ShiftRows — en este paso se realiza una transposición donde cada fila del estado es rotada de manera cíclica un número determinado de veces.

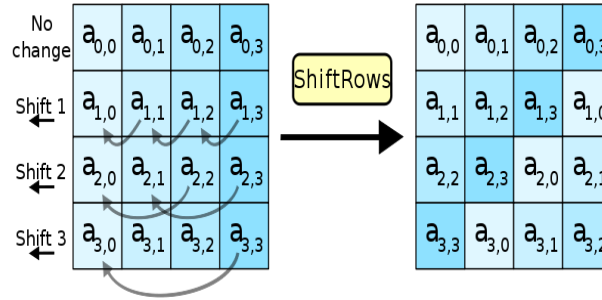


Figura 1.2. Fase de ShiftRows del cifrado AES.

3. MixColumns — se mezclan las columnas del estado, combinando los cuatro bytes en cada columna usando una transformación lineal.

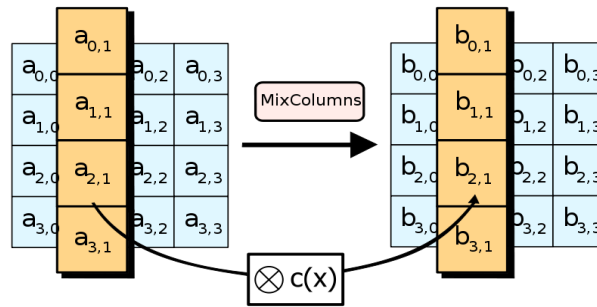


Figura 1.3. Fase de MixColumns del cifrado AES.

4. AddRoundKey — cada byte del estado es combinado con la clave de ronda; cada clave de ronda se deriva de la clave de cifrado usando una iteración de la clave.

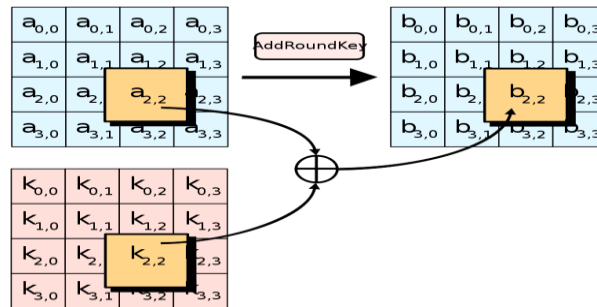


Figura 1.4. Fase de AddRoundKey del cifrado AES

- Etapa final:
 1. SubBytes
 2. ShiftRows
 3. AddRoundKey

Descripción

- SubBytes

Cada byte en la matriz es actualizado usando la caja-S de Rijndael de 8 bits. Esta operación provee la no linealidad en el cifrado. Para evitar ataques basados en simples propiedades algebraicas, la caja-S se construye por la combinación de la función inversa con una transformación afín invertible.

- ShiftRows

Opera en las filas del estado; rota de manera cíclica los bytes en cada fila por un determinado offset. En AES, la primera fila queda en la misma posición. Cada byte de la segunda fila es rotado una posición a la izquierda. De manera similar, la tercera y cuarta filas son rotadas por los offsets de dos y tres respectivamente. De esta manera, cada columna del estado resultante del paso ShiftRows está compuesta por bytes de cada columna del estado inicial.

- MixColumns

Cada columna se trata como un polinomio y luego se multiplica el módulo $x^4 + 1$ con un polinomio fijo $C(x)$. El paso MixColumns puede verse como una multiplicación matricial en el campo finito de Rijndael.

- AddRoundKey

En este paso la subclave se combina con el estado. En cada ronda se obtiene una subclave de la clave principal, usando la iteración de la clave; cada subclave es del mismo tamaño que el estado. La subclave se agrega combinando cada byte del estado con el correspondiente byte de la subclave usando XOR.

Ingeniería de software

La ingeniería de software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. El proceso de la ingeniería de software es el elemento que mantiene juntos los estratos de la tecnología y que permite el desarrollo racional y a tiempo del software de computadora. El proceso define un marco de trabajo que debe establecerse para la entrega efectiva de la tecnología de la ingeniería de software.

Los métodos de la ingeniería de software proporcionan los procedimientos técnicos para construir software, abarcan un amplio espectro de tareas que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte. Los métodos se basan en un conjunto de principios básicos que gobiernan cada área de la tecnología e incluye actividades de modelado y otras técnicas descriptivas.

Marco de trabajo para el proceso

Un marco de trabajo establece la base para un proceso de software completo al identificar las actividades de este que se aplican a todos los proyectos de software, independientemente del tamaño y la complejidad de estos.

El siguiente es un marco de trabajo genérico de proceso que puede aplicarse en la mayoría de los proyectos de software:

- **Comunicación:** esta actividad implica una intensa colaboración y comunicación con los clientes; además, abarca la investigación de requisitos y otras actividades relacionadas.
- **Planeación:** establece un plan para el trabajo de la ingeniería del software. Describe las tareas técnicas que deben realizarse, los riesgos probables, los recursos que serán requeridos, los productos del trabajo que han de producirse y un programa de trabajo.
- **Modelado:** abarca la creación de modelados que permiten al desarrollador y al cliente entender mejor los requisitos del software y el diseño que lograra satisfacerlos.
- **Construcción:** combina la generación del código y la realización de pruebas necesarias para descubrir errores en el código.
- **Despliegue:** el software se entrega al cliente quien evalúa el producto recibido y proporciona información basada en su evaluación.

A esta visión general de la ingeniería de software la completa una serie de actividades de apoyo que incluyen:

- Seguimiento y control del proyecto de software: permite que el equipo de software evalúe el progreso comparándolo con el plan del proyecto y así tomar acciones necesarias para mantener el programa.
- Gestión de riesgo: evalúa los riesgos que pudieran afectar los resultados de proyecto o la calidad del producto.
- Aseguramiento de la calidad del software: define y conduce las actividades requeridas para asegurar la calidad del software.
- Revisiones técnicas formales: evalúa los productos del trabajo de la ingeniería del software en un esfuerzo encaminado a descubrir y eliminar los errores antes de que estos se propaguen hacia la siguiente acción o actividad.
- Medición: define y recolecta mediciones del proceso, el proyecto y el producto para ayudar al equipo a entregar software que satisfaga las necesidades del cliente; se puede usar en conjunto con todas las otras actividades del marco de trabajo.
- Gestión de la configuración del software: maneja los efectos del cambio a través del proceso del software.
- Gestión de la reutilización: define los criterios para la reutilización de productos del trabajo y establece mecanismos para la creación de componentes reutilizables.
- Preparación y producción del producto de trabajo: abarca las actividades requeridas para crear productos del trabajo como modelos, documentos, registros, formatos y listas.

Lenguajes de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar órdenes que pueden ser ejecutadas por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

La programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

- El desarrollo lógico del programa para resolver un problema en particular.
- Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa).
- Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina.
- Prueba y depuración del programa.
- Desarrollo de la documentación.

Permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Una característica relevante de los lenguajes de programación es precisamente que más de un programador pueda usar un conjunto común de instrucciones que sean comprendidas entre ellos para realizar la construcción de un programa de forma colaborativa.

Paradigmas de programación

Un paradigma es la manera como percibimos el mundo y nos ayuda a predecir su comportamiento.

Es una forma aceptada de resolver un problema en la ciencia, que más tarde es utilizada como modelo para la investigación y la formación de una teoría. También se refiere al conjunto de métodos, reglas y generalizaciones utilizadas conjuntamente por aquellos entrenados para realizar el trabajo científico de investigación.

Así, un paradigma de programación nos indican las diversas formas que, a lo largo de la evolución de los lenguajes, han sido aceptadas como estilos para programar y para resolver los problemas por medio de una computadora.

Programación orientada a objetos

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

Los objetos son entidades que tienen un determinado estado, comportamiento (método) e identidad:

- El estado está compuesto de datos, será uno o varios atributos a los que se habrán asignado unos valores concretos (datos).
- El comportamiento está definido por los métodos o mensajes a los que sabe responder dicho objeto, es decir, qué operaciones se pueden realizar con él.
- La identidad es una propiedad de un objeto que lo diferencia del resto, dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante).

Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos. A su vez, los objetos disponen de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos. Esta comunicación favorece a su vez el cambio de estado en los propios objetos.

Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separa el estado y el comportamiento.

Los métodos (comportamiento) y atributos (estado) están estrechamente relacionados por la propiedad de conjunto. Esta propiedad destaca que una clase requiere de métodos para poder tratar los atributos con los que cuenta. El programador debe pensar indistintamente en ambos conceptos, sin separar ni darle mayor importancia a alguno de ellos. Hacerlo podría producir el hábito erróneo de crear clases contenedoras de información por un lado y clases con métodos que manejen a las primeras por el otro. De esta manera se estaría realizando una programación estructurada camuflada en un lenguaje de programación orientado a objetos.

La POO difiere de la programación estructurada tradicional, en la que los datos y los procedimientos están separados y sin relación, ya que lo único que se busca es el procesamiento de unos datos de entrada para obtener otros de salida.

Características de la POO

- **Abstracción:** denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.
- **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema.
- **Modularidad:** Permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos.
- **Principio de ocultación:** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas.
- **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.
- **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.
- **Recolección de basura:** la recolección de basura o garbage collector es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando.

Hojas de estilo en cascada

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML.

Sintaxis

CSS tiene una sintaxis muy sencilla, que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos.

Una hoja de estilos CSS consiste en una serie de reglas. Cada regla consiste en uno o más selectores y un bloque de estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor;

Los selectores pueden afectar a uno o varios elementos a la vez, en función de su tipo, nombre (name), ID, clase (class), etcétera

```
selector [, selector2,...] [: pseudo-class] {  
    propiedad: valor;  
    [propiedad2: valor2;  
    ...]  
}  
/* comentarios */
```

Para dar formato a un documento HTML, puede emplearse CSS de tres formas distintas, mediante CSS introducido por el autor del HTML, una hoja de estilo interna o una hoja de estilo externa.

Ventajas

Algunas ventajas de utilizar CSS son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.
- Optimización del ancho de banda de la conexión, pues pueden definirse los mismos estilos para muchos elementos con un sólo selector; o porque un mismo archivo CSS puede servir para una multitud de documentos.
- Mejora en la accesibilidad del documento, pues con el uso del CSS se evitan antiguas prácticas necesarias para el control del diseño (como las tablas), y que iban en perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales.

1.2 Información

De forma general, las aplicaciones web que se desarrollan dentro de cualquier organización implementan algún sistema de autenticación de usuarios.

Esto implica que cada usuario tenga una contraseña y como tal, un espacio en la infraestructura lógica de la red, ya sea en una base de datos o en un archivo. Sin embargo, existen usuarios que tienen acceso a más de una aplicación y por lo tanto más de una contraseña, lo que incrementa el espacio necesario para cada una sin mencionar lo complejo que se vuelve para el usuario recordarlas todas.

Además, no siempre se lleva un registro de las aplicaciones desarrolladas o de los usuarios y contraseñas de cada una, es decir, cada aplicación cuenta con una base de datos independiente de las demás, complicando así la administración de toda la infraestructura.

1.3 Metodología y estrategia de desarrollo

Toda organización, dependencia o instituto que se dedique o cuente con un departamento de TI y/o Ingeniería de software, debe describir un conjunto único de actividades dentro de su marco de trabajo para el proceso de software que adopte.

Además, debe adaptar el modelo de proceso resultante y ajustarlo a la naturaleza específica de cada proyecto, a las personas que lo realizarán, y el ambiente en el que se ejecutara el trabajo.

De manera general y tradicional, se ha establecido un marco de trabajo genérico para el proceso, el cual incluye las siguientes actividades: comunicación, planeación, modelado, construcción y desarrollo.

Existen diferentes modelos del proceso de software que prescriben un conjunto de elementos y un flujo de trabajo que es la forma en la que estos elementos del proceso se interrelacionan ente sí.

Para el desarrollo del Sistema Integral de Administración y Control de Usuarios Online se implementa el tradicional Modelo en Cascada.

Modelo en cascada

Algunas veces llamado el Ciclo de Vida Clásico, sugiere un enfoque sistemático, secuencial hacia el desarrollo del software, que se inicia con la especificación de requerimientos del cliente y que continúa con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del software terminado (Figura 1.1).

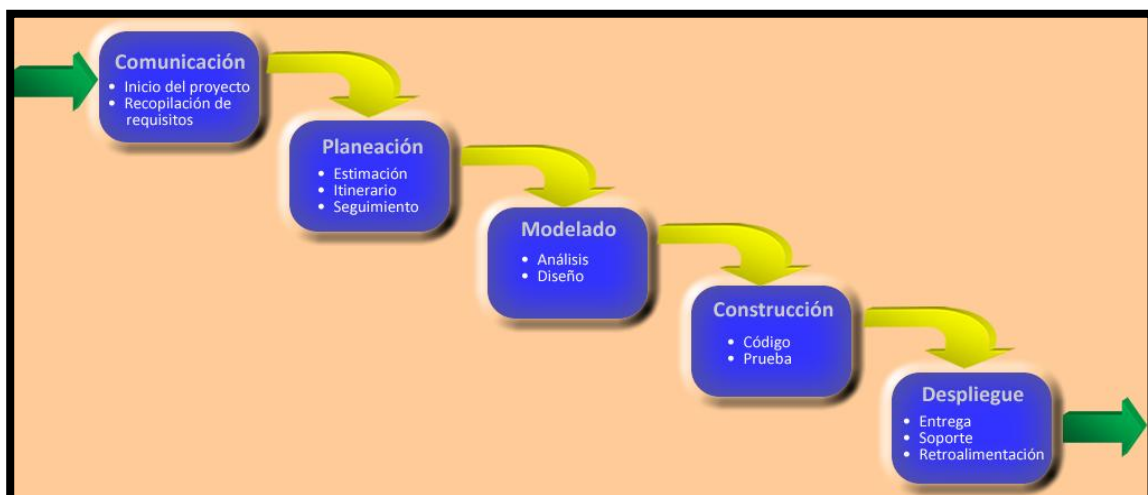


Fig. 1.5. Representación del modelo en cascada, el modelo original propuesto por Winston Royce prevé ciclos de retroalimentación.

1.4 Herramientas de desarrollo

JDeveloper 10.1.4

Es un entorno de desarrollo proporcionado por Oracle. Ofrece características para desarrollo en Java, XML, SQL and PL/SQL, HTML, JavaScript, BPEL y PHP. Con este software, Oracle busca simplificar el desarrollo de aplicaciones enfocándose en proveer un acercamiento visual y declarativo junto con un ambiente de codificación avanzado.

MySQL

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Está desarrollado en su mayor parte en ANSI C. Es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento de MySQL. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

Bizagi Process Modeler

Es un software libre para diagramar y documentar procesos de manera gráfica en un formato estándar conocido como BPMN (Business Process Modeling Notation). Su "comportamiento inteligente" permite a los usuarios no-técnicos diagramar procesos arrastrando las figuras hacia el sitio adecuado en el diagrama.

Una vez se han creado los diagramas de proceso y la documentación correspondiente, éstos pueden exportarse a Word, PDF, Visio, la web o SharePoint5 para compartirlos con otras personas que no tengan el modelador de procesos instalado.

Bizagi sigue un modelo de "no código" que permite crear aplicaciones basadas en procesos, coordinando gente y sistemas sin la necesidad de programación. Bizagi puede ser utilizado para diagramar y automatizar cualquier tipo de proceso, desde el más simple hasta el más complejo.

Dia

Dia es una aplicación informática de propósito general para la creación de diagramas, desarrollada como parte del proyecto GNOME. Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades. Está diseñado como un sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas. Actualmente se incluyen diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, diagramas de circuitos eléctricos, etc. Nuevas formas pueden ser fácilmente agregadas, dibujándolas con un subconjunto de SVG e incluyéndolas en un archivo XML.

El formato para leer y almacenar gráficos es XML (comprimido con gzip, para ahorrar espacio). Puede producir salida en los formatos EPS, SVG y PNG. También conviene recordar que Dia, gracias al paquete dia2code, puede generar el esqueleto del código a escribir, si utilizáramos con tal fin un UML.

1.5 Estado del arte. Alternativas de solución

Single Sign-On

Es un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con un solo identificador.

Hay cinco tipos principales de SSO:

- **Enterprise single sign-on (E-SSO)**, también llamado legacy single sign-on, funciona para una autenticación primaria, interceptando los requerimientos de login presentados por las aplicaciones secundarias para completar los mismos con el usuario y contraseña. Los sistemas E-SSO permiten interactuar con sistemas que pueden deshabilitar la presentación de la pantalla de login.
- **Web single sign-on (Web-SSO)**, también llamado Web access management (Web-AM) trabaja sólo con aplicaciones y recursos accedidos vía web. Los accesos son interceptados con la ayuda de un servidor proxy o de un componente instalado en el servidor web destino. Los usuarios no autenticados que tratan de acceder son redirigidos a un servidor de autenticación y regresan solo después de haber logrado un acceso exitoso. Se utilizan cookies, para reconocer aquellos usuarios que acceden y su estado de autenticación.
- **Kerberos** es un método popular de externalizar la autenticación de los usuarios. Los usuarios se registran en el servidor Kerberos y reciben un "ticket", luego las aplicaciones-cliente lo presentan para obtener acceso.

- **Identidad federada** es una nueva manera de concebir este tema, también para aplicaciones Web. Utiliza protocolos basados en estándares para habilitar que las aplicaciones puedan identificar los clientes sin necesidad de autenticación redundante.
- **OpenID** es un proceso de SSO distribuido y descentralizado donde la identidad se compila en una url que cualquier aplicación o servidor puede verificar.

En la tabla 1.1 se enlistan algunas implementaciones del procedimiento de Single Sign-On.

Nombre	Proyecto/Distribuidor	Tipo
Windows Live ID	Microsoft	Comercial
Ubuntu Single Sign On	Canonical Ltd.	Comercial
Kerberos	M.I.T.	Protocolo
Facebook connect	Facebook	Específico para Facebook

Tabla 1.1 Implementaciones de Single Sign-On

CAPÍTULO 2
DISEÑO DEL SISTEMA

2.1 Requerimientos

Para el SIAC-Online es necesario contar con la Máquina Virtual de Java (JVM) que es el intérprete que ejecuta el programa y es la clave de muchas de las características principales de este, como la portabilidad, la eficiencia y la seguridad.

Debe permitirse la ejecución de JavaScript en el explorador de internet que se use habitualmente porque, a través de este lenguaje, se realizan las validaciones necesarias para que el sistema continúe con su flujo normal.

Es necesario también contar con un programa que lea archivos pdf para visualizar correctamente los documentos generados por el sistema.

Cuando el sistema envía un correo o consulta la base de datos para verificar y comparar contraseñas, nombres de usuario, sistemas y roles, se realiza una conexión a través de la red y es por esto que el usuario debe contar con una conexión a internet.

2.2 Alcance

Los usuarios principales del Sistema Integral de Administración y Control de Usuarios Online serán aquellos que trabajen directa o indirectamente con la organización, es decir, quienes requieran manipular la información o los servicios que esta ofrece.

2.3 Proceso

El proceso que se lleva a cabo con el SIAC-Online para el registro, la autorización o cancelación de usuarios se muestra en el diagrama BPM de la figura 2.1.

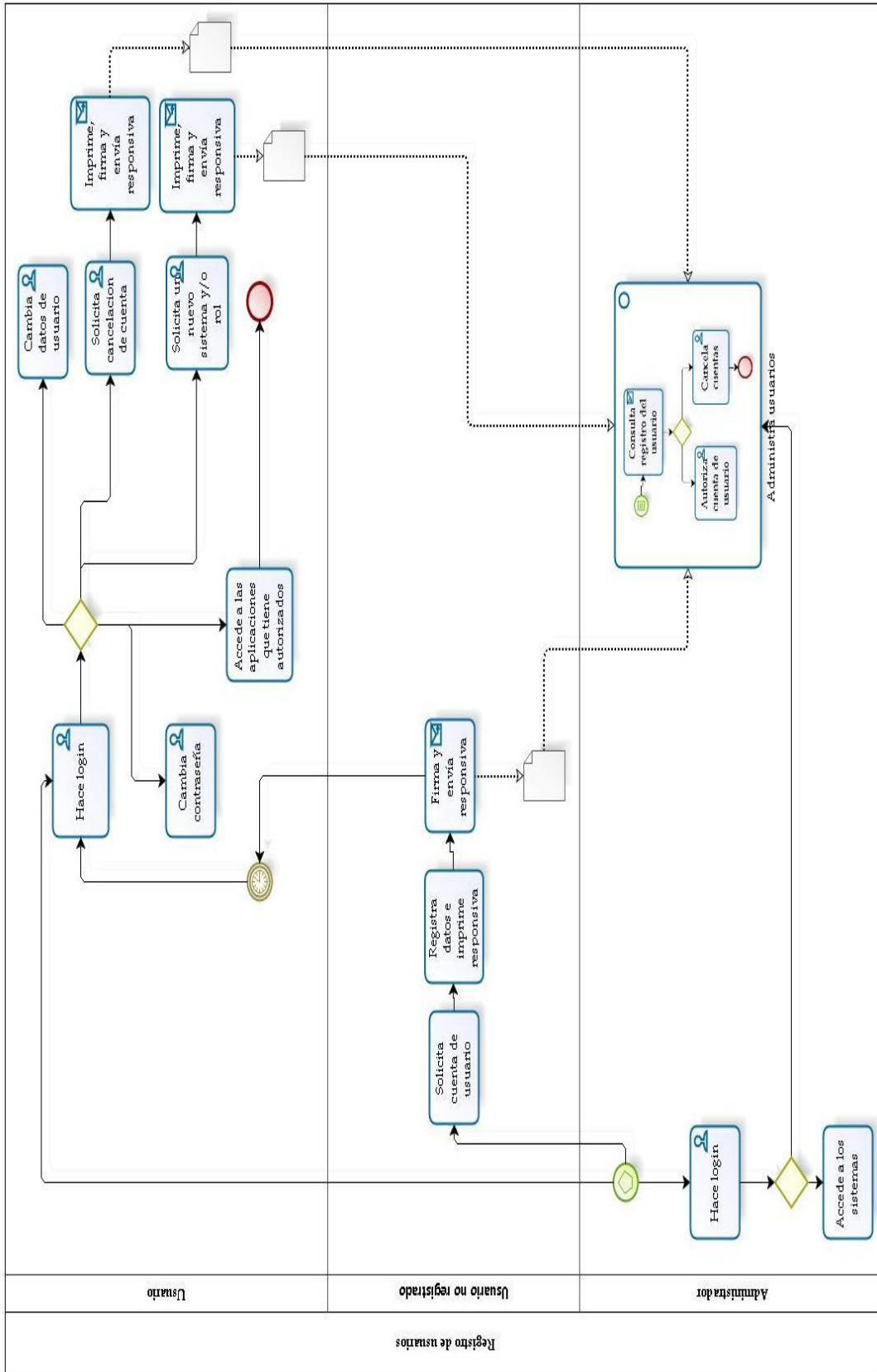


Figura 2.1 Flujo de procesos del sistema.

2.4 Casos de uso

Diagrama

En la figura 2.2 se representan los casos de uso del sistema en UML.

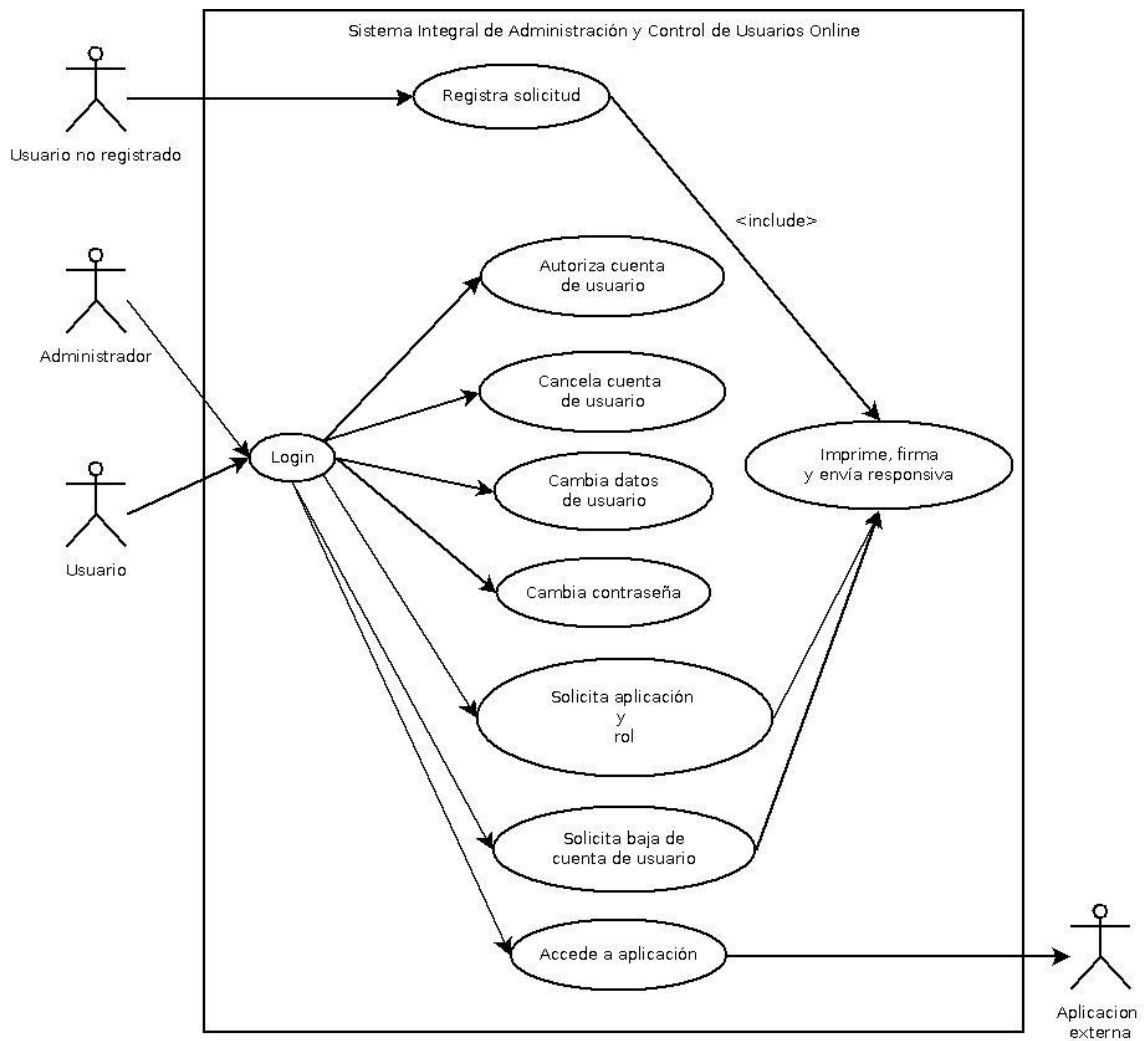


Figura 2.2 Diagrama de casos de uso del sistema.

Realización de casos de uso

Se describen las interacciones directas que tiene el sistema con el usuario y con otros sistemas o aplicaciones, es decir, se detalla paso a paso el diagrama de casos de uso de la figura 2.2.

Caso de Uso	Registra solicitud	
Objetivo	Crear un nuevo registro de usuario en el sistema.	
Descripción	La primera vez que una persona desea hacer uso de los servicios del sistema necesita registrarse. Para ello el sistema solicitará los datos necesarios del usuario. El registro será unívoco.	
Actores	Usuario no registrado	
	Paso	Acción
Precondiciones	1.	El sistema debe estar disponible. Debe ser la primera vez que el usuario accede al sistema por completo.
Postcondiciones	1.	El nuevo usuario queda registrado en el sistema.
Secuencia Normal	1.	Un usuario no registrado accede al sistema.
	2.	El sistema despliega la pantalla de login para la autenticación del usuario.
	3.	El nuevo usuario solicita su registro.
	4.	El sistema solicita información para crear el registro.
	5.	El usuario ingresa información al sistema y envía sus datos.
	6.	El sistema valida la información y crea el nuevo registro, notificándolo al usuario.
	7.	El sistema genera la responsiva correspondiente.

	8.	El usuario ya registrado imprime, firma y envía la responsiva al administrador.
Excepciones	1.	Los datos introducidos por el usuario no son correctos, por usar caracteres no válidos o por dejar campos obligatorios vacíos, etc. El sistema volverá a solicitar los datos.
	2.	El usuario abandona el sistema (o la aplicación se cierra por cualquier razón) sin terminar de definir la información necesaria para el registro.

Caso de Uso	Login	
Objetivo	Autenticar al usuario.	
Descripción	Un usuario registrado ingresa al sistema y es autenticado para acceder a las aplicaciones que tiene asignadas o para actualizar su perfil dentro del sistema.	
Actores.	Usuario registrado	
	Paso	Acción
Precondiciones	1.	Deberá existir el usuario registrado y habilitado por el Administrador.
Postcondiciones	1.	El usuario hace uso del sistema y puede acceder a las diferentes aplicaciones que tiene asignadas.
Secuencia Normal	1.	El usuario registrado ingresa usuario y contraseña (login).
	2.	El sistema autentica al usuario y permite su inicio de sesión.
Excepciones	1.	NO existe el usuario o la contraseña es incorrecta. El sistema envía mensaje de error y solicita nuevamente los datos.

Caso de Uso	Autoriza cuenta de usuario	
Objetivo	Habilitar el acceso de las cuentas de usuario.	
Descripción	Una vez que se recibe la responsiva de cuenta de usuario debidamente firmada, el administrador habilita al usuario para que acceda a su perfil dentro del sistema.	
Actores	El administrador del sistema	
	Paso	Acción
	1.	El administrador tiene sesión activa en el sistema.
Precondiciones	2.	El usuario está registrado.
	3.	Se ha recibido la responsiva debidamente firmada.
	4.	Existe el registro de la solicitud en el sistema.
Postcondiciones	1.	El usuario está habilitado para acceder a las aplicaciones.
Secuencia Normal	1.	El administrador recibe la responsiva de cuenta de usuario debidamente firmada.
	2.	El administrador consulta la información del usuario.
	3.	El administrador habilita al usuario para acceder al sistema bajo el rol que fue solicitado.
	4.	El sistema confirma la acción cambiando el estatus del usuario.
Excepciones	1.	No se ha recibido la responsiva de cuenta de usuario o no está debidamente firmada.
	2.	La aplicación o rol solicitados no corresponden a los esperados.
	3.	No existe el registro de la solicitud en el sistema.

Caso de Uso	Cancela cuenta de usuario	
Descripción	El administrador del sistema elimina el registro de un usuario previa solicitud escrita de este.	
Objetivo	Eliminar un registro de un usuario del sistema.	
Actores	Administrador del sistema	
	Paso	Acción
Precondiciones	1.	El administrador tiene sesión activa en el sistema.
	2.	Debe de existir la solicitud de baja debidamente firmada.
Postcondiciones	1.	El registro de un usuario del sistema es eliminado por completo.
Secuencia Normal	1.	El administrador recibe solicitud de baja
	2.	Consulta al usuario y elimina el registro.
	3.	El sistema notifica la eliminación del registro.
Extensiones	1.	<p>Si el usuario es un administrador de sistema:</p> <ul style="list-style-type: none"> • El administrador desea dar de baja a otro usuario. • El sistema comprueba los permisos del usuario. • El sistema permite elegir el usuario a dar de baja. • El administrador elige el usuario. • El sistema da de baja al usuario seleccionado.
	2.	<p>En cualquiera de las secuencias anteriores:</p> <ul style="list-style-type: none"> • Si la comprobación de los permisos no es satisfactoria, el sistema no permite realizar la operación.

Caso de Uso	Cambia datos de usuario	
Objetivo	Modificar un perfil definido para un usuario.	
Descripción	Un usuario registrado en el sistema inicia sesión para modificar sus datos.	
Actores	Usuario registrado	
	Paso	Acción
Precondiciones	1.	Debe existir el usuario y el sistema debe estar disponible.
Postcondiciones	1.	La información correspondiente a este usuario está actualizada.
Secuencia Normal	1.	El usuario hace login en el sistema.
	2.	El usuario indica que desea actualizar sus datos.
	3.	El sistema muestra al usuario los datos que puede modificar.
	4.	El usuario ingresa los datos que desea modificar y envía la información actualizada.
	5.	El sistema valida y guarda la información.
	6.	El sistema notifica al usuario del cambio realizado.
Excepciones	1.	Los datos modificados no son correctos o son inválidos. Debe notificarse al usuario y permitirle rectificar.
	2.	El usuario puede cancelar la acción volviendo al instante inicial y sin modificarse información alguna del perfil.

Caso de Uso	Cambia contraseña	
Objetivo	Actualizar la contraseña del usuario	
Descripción	El usuario registrado realiza el cambio de su contraseña cuando así lo requiera.	
Actores	Usuario registrado	
	Paso	Acción
Precondiciones	1.	Ser un usuario registrado.
	2.	Sesión iniciada dentro del sistema.
Postcondiciones	1.	Usuario con contraseña actualizada
Secuencia Normal	1.	El usuario entra al sistema y selecciona la opción de cambio de contraseña.
	2.	Ingresa la contraseña anterior.
	3.	Ingresa la nueva contraseña.
	4.	Confirma la nueva contraseña.
	5.	Envía la información actualizada.
	6.	El sistema valida la nueva contraseña.
	7.	El sistema confirma el cambio de contraseña.
Excepciones	1.	La nueva contraseña y su confirmación no coinciden. El sistema muestra un mensaje de error y el usuario ingresa nuevamente los datos.
	2.	La nueva contraseña tiene menos de 6 caracteres. El sistema muestra un mensaje de error y el usuario ingresa nuevamente los datos.

Caso de Uso	Solicita aplicación y rol	
Objetivo	Solicitar el acceso a una aplicación mediante un rol.	
Descripción	El usuario registrado solicita un rol dentro de una aplicación que esté disponible y que le sea necesaria.	
Actores	Usuario registrado	
	Paso	Acción
Precondiciones	1.	Ser un usuario registrado.
	2.	Sesión iniciada dentro del sistema.
Postcondiciones	1.	La solicitud de acceso a la aplicación está registrada en el sistema.
Secuencia Normal	1.	El usuario entra al sistema y selecciona la opción de solicitar un rol dentro de una aplicación.
	2.	Selecciona la aplicación que necesite.
	3.	Selecciona el rol que requiere dentro de la aplicación.
	4.	Envía la información de la solicitud para el registro.
	5.	El sistema confirma el registro y muestra la opción de impresión de responsiva.
	6.	El usuario imprime y firma la responsiva para enviarla al administrador.
Excepciones	1.	El usuario ya está registrado en la aplicación y rol solicitados. El sistema muestra mensaje de error.
Requerimientos No Funcionales	Es necesario que la responsiva sea enviada debidamente firmada al administrador para que este pueda habilitar al usuario dentro de la aplicación solicitada.	

Caso de Uso	Solicita baja de cuenta de usuario.	
Descripción	El usuario registrado solicita la baja de su cuenta dentro del sistema.	
Objetivo	Registrar la solicitud de baja de la cuenta del usuario.	
Actores	Usuario registrado	
	Paso	Acción
Precondiciones	1.	Ser un usuario registrado.
	2.	Sesión iniciada dentro del sistema.
Postcondiciones	1.	La solicitud de baja de la cuenta de usuario está registrada.
Secuencia Normal	1.	El usuario accede al sistema y selecciona la opción de solicitud de baja.
	2.	El sistema muestra información del usuario y solicita un motivo de baja.
	3.	El usuario selecciona el motivo de su solicitud de baja.
	4.	El usuario ingresa su contraseña para confirmar la solicitud de baja.
	5.	El sistema muestra una advertencia y solicita de nuevo la confirmación de la solicitud de baja.
	6.	El sistema confirma el registro de la solicitud de baja del usuario.
Excepciones	1.	La contraseña del usuario es incorrecta. El sistema muestra mensaje de error y solicita nuevamente los datos.

Caso de Uso	Accede a aplicación.	
Objetivo	Permitir al usuario acceder a las aplicaciones que tiene asignadas.	
Descripción	Una vez que el sistema autentica al usuario, le presentará las aplicaciones a las que puede tener acceso bajo el rol que tiene asignado. El usuario seleccionará la que utilizará en ese momento.	
Actores	Usuario Registrado	
	Paso	Acción
Precondiciones	1.	Ser un usuario registrado.
	2.	Sesión iniciada dentro del sistema.
	3.	La aplicación está disponible.
	4.	El usuario está habilitado dentro del sistema para acceder a la aplicación.
Postcondiciones	1.	El usuario accede a la aplicación deseada.
Secuencia Normal	1.	El usuario entra al sistema y selecciona la opción de selección de aplicación.
	2.	El sistema despliega las aplicaciones a las que el usuario tiene acceso.
	3.	El usuario selecciona la aplicación deseada.

2.5 Diagrama entidad-relación

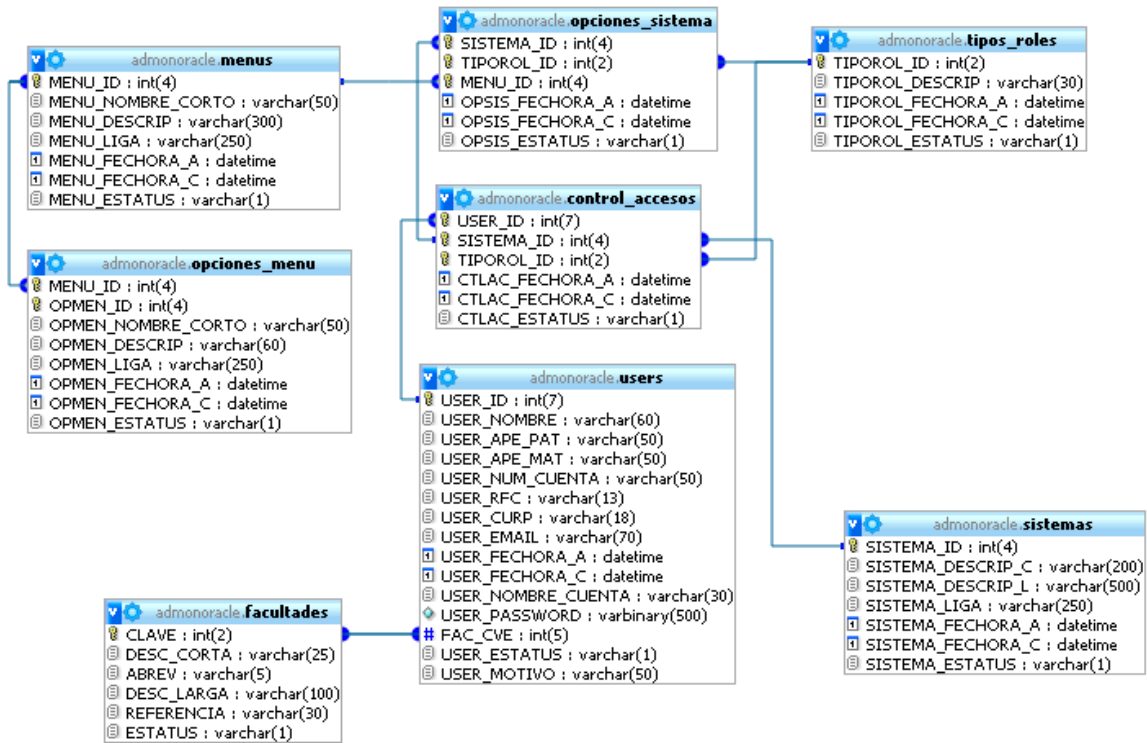


Figura 2.3 Diagrama E-R de la base de datos del sistema.

2.6 Diccionario de datos

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en la base de datos del sistema que se programa.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema.

RELACIÓN DE TABLAS

CONSEQ.	TABLA
1	USERS
2	SISTEMAS
3	TIPOS_ROLES
4	CONTROL_ACCESOS
5	OPCIONES_SISTEMA
6	MENUS
7	OPCIONES_MENU
8	FACULTADES

Tabla 2.1 Relación de las tablas utilizadas en el sistema.

a) **Tabla USERS**

Tabla primaria que contiene el registro de toda la información relevante del usuario.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	USER_ID	INT	5	Identificador de usuario.
2	USER_NOMBRE	VARCHAR	60	Nombre o nombres propios del usuario.
3	USER_APE_PAT	VARCHAR	50	Apellido paterno del usuario.
4	USER_APE_MAT	VARCHAR	50	Apellido materno del usuario.
5	USER_NUM_CUENTA	VARCHAR	50	Número de cuenta del usuario.
6	USER_RFC	VARCHAR	13	RFC del usuario, puede ser con o sin homoclave.
7	USER_CURP	VARCHAR	18	CURP del usuario.
8	USER_EMAIL	VARCHAR	70	Correo electrónico del usuario.
9	USER_FECHORA_A	DATETIME		Fecha y hora de alta del usuario.
10	USER_FECHORA_C	DATETIME		Fecha y hora de cualquier cambio en el registro.
11	USER_NOMBRE_CUENTA	VARCHAR	30	Nombre de cuenta del usuario. Este será su identificador dentro del sistema.

12	USER_PASSWORD	VARBINARY	500	Contraseña del usuario. Se almacena cifrada con AES.
13	FAC_CVE	INT	2	Clave de la facultad a la que pertenece el usuario. Es llave foránea.
14	USER_ESTATUS	VARCHAR	1	Estatus actual del usuario. (V)igente o (C)ancelado.
15	USER_MOTIVO	VARCHAR	50	Motivo por el que se cancela el usuario.

Tabla 2.2 Detalle de tabla USERS.

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	USER_ID	Llave primaria conformada por el identificador del usuario.
2	FAC_CVE	FAC_CVE	Llave foránea de la tabla USERS que la relaciona con la tabla FACULTADES.

Tabla 2.3 Llaves primarias y foráneas de la tabla USERS.

b) **Tabla SISTEMAS**

Tabla que contiene la información de los sistemas disponibles.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	SISTEMA_ID	INT	4	Identificador del sistema.
2	SISTEMA_DESCRIP_C	VARCHAR	200	Descripción corta del sistema que se utiliza en pantallas.
3	SISTEMA_DESCRIP_L	VARCHAR	500	Descripción larga del sistema. Es el nombre oficial del sistema.
4	SISTEMA_LIGA	VARCHAR	250	Liga del sistema accesible desde la red.
5	SISTEMA_FECHORA_A	DATETIME		Fecha y hora en que se da de alta el sistema.
6	SISTEMA_ESTATUS	VARCHAR	1	Estatus del sistema.

Tabla 2.4 Detalle de tabla SISTEMAS.

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	SISTEMA_ID	Llave primaria conformada por el identificador del sistema.

Tabla 2.5 Llaves primarias y foráneas de la tabla SISTEMAS.

c) **Tabla TIPOS_ROLES**

Catálogo que contiene los roles disponibles y sus descripciones para que sean habilitados en cada sistema, de acuerdo a las especificaciones de éstos.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	TIPOROL_ID	INT	2	Identificador del rol.
2	TIPOROL_DESCRIP	VARCHAR	30	Descripción o nombre del rol.
3	TIPOROL_FECHORA_A	DATETIME		Fecha y hora en que se da de alta el rol.
4	TIPOROL_ESTATUS	VARCHAR	1	Estatus del rol.

Tabla 2.6 Catálogo de roles disponibles.

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	TIPOROL_ID	Llave primaria conformada por el identificador del rol.

Tabla 2.7 Llaves primarias y foráneas de la tabla TIPOS_ROLES.

d) Tabla CONTROL_ACCESOS

En esta tabla se almacenan los identificadores de usuario, de sistema y de rol. Es a través de esta tabla que se controla el inicio de sesión de los usuarios.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	USER_ID	INT	5	Identificador del usuario.
2	SISTEMA_ID	INT	4	Identificador del sistema.
3	TIPOROL_ID	INT	2	Identificador del rol asignado.
4	CTLAC_FECHORA_A	DATETIME		Fecha y hora en que se da de alta el acceso.
5	CTLAC_FECHORA_C	DATETIME		Fecha y hora en que se hace un cambio en el registro.
6	CTLAC_ESTATUS	VARCHAR	1	Estatus del acceso.

Tabla 2.8 Detalle de tabla CONTROL_ACCESOS.

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	USER_ID SISTEMA_ID TIPOROL_ID	Llave primaria conformada por el identificador del usuario, el identificador del sistema y el identificador del rol.

Tabla 2.9 Llaves primarias y foráneas de la tabla CONTROL_ACCESOS.

e) **Tabla OPCIONES_SISTEMA**

Tabla que contiene los roles que cada sistema tiene habilitados así como el menú que corresponde a cada rol.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	SISTEMA_ID	INT	4	Identificador del sistema.
2	TIPOROL_ID	INT	2	Identificador del rol asignado.
3	MENU_ID	INT	4	Identificador del menú.
4	OP SIS_FECHORA_A	DATETIME		Fecha y hora en que se da de alta la opción.
5	OP SIS_ESTATUS	VARCHAR	1	Estatus de la opción del menú.

Tabla 2.10 Detalle de tabla OPCIONES_SISTEMA

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	SISTEMA_ID TIPOROL_ID MENU_ID	Llave primaria conformada por el identificador del sistema, el identificador del rol y el identificador del menú.

Tabla 2.11 Llaves primarias y foráneas de la tabla OPCIONES_SISTEMA.

f) **Tabla MENUS**

Tabla donde se almacenan los menus disponibles, éstos se habilitan para cada sistema a través de la tabla OPCIONES_SISTEMA.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	MENU_ID	INT	4	Identificador del menú.
1	MENU_NOMBRE_CORTO	VARCHAR	50	Nombre abreviado del menú.
2	MENU_DESCRIP	VARCHAR	300	Descripción detallada del menú.
3	MENU_LIGA	VARCHAR	250	Liga o vínculo del menú dentro del servidor.
4	MENU_FECHORA_A	DATETIME		Fecha y hora en que se da de alta el menú.
5	MENU_ESTATUS	VARCHAR	1	Estatus del menú.

Tabla 2.12 Detalle de tabla MENUS.

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	MENU_ID	Llave primaria conformada por el identificador del menú.

Tabla 2.13 Llaves primarias y foráneas de la tabla MENUS.

g) **Tabla OPCIONES_MENU**

Esta tabla almacena las opciones que tiene disponibles cada menú.

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	MENU_ID	INT	4	Identificador del menú.
1	OPMEN_ID	INT	4	Identificador de la opción.
2	OPMEN_NOMBRE_CORTO	VARCHAR	50	Nombre abreviado de la opción que se utiliza en pantalla.
3	OPMEN_DESCRIP	VARCHAR	100	Descripción detallada de la opción.
4	OPMEN_LIGA	VARCHAR	250	Liga o vínculo de la opción dentro del servidor de aplicaciones.
5	OPMEN_FECHORA_A	DATETIME		Fecha y hora en que se da de alta la opción.
6	OPMEN_ESTATUS	VARCHAR	1	Estatus de la opción.

Tabla 2.14 Detalle de tabla OPCIONES_MENU.

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	MENU_ID, OPMEN_ID	Llave primaria conformada por el identificador del menú y el identificador de la opción.

Tabla 2.15 Llaves primarias y foráneas de la tabla OPCIONES_MENU.

h) **Tabla FACULTADES**

Catálogo de las facultades y escuelas de la UNAM

Detalle				
#	Nombre del campo	Tipo	Long.	Descripción
1	FACULTAD_CLAVE	INT	2	Clave de la facultad o escuela.
2	FACULTAD_ABREV	VARCHAR	5	Abreviación del nombre de la escuela o facultad.
3	FACULTAD_DESC_C	VARCHAR	25	Descripción corta de la facultad o escuela.
4	FACULTAD_DESC_L	VARCHAR	100	Nombre completo de la escuela o facultad
5	FACULTAD_REFERENCIA	VARCHAR	30	Nombre de referencia de la facultad o escuela
6	FACULTAD_FECHORA_A	DATETIME		Fecha y hora en que se registra la facultad en el catalogo.
7	FACULTAD_ESTATUS	VARCHAR	1	Estatus de la facultad

Tabla 2.16 Detalle de tabla FACULTADES

Llaves primarias, foráneas e índices			
#	Nombre	Componentes	Descripción
1	PRIMARY	CLAVE	Llave primaria conformada por la clave de la escuela o facultad.

Tabla 2.17 Llaves primarias y foráneas de la tabla FACULTADES.

CAPÍTULO 3
DESARROLLO DEL SISTEMA

En este capítulo se describen las clases y librerías de Java implementadas en el desarrollo del Sistema Integral de Administración y Control de Usuarios Online así como el flujo del proceso entre cada uno de sus módulos.

En la figura 3.1 se listan las clases y las librerías utilizadas así como la página JSP que las implementa.

Clase o librería	Java Server Page (JSP)
Java DataBase Connectivity (JDBC) ManejaBD.java ConsultaBD.java	login.jsp ConsultaRegistro.jsp
iText ResponsivaPDF.java	Responsiva.jsp ConsultaRegistro.jsp AdminLogin.jsp UserLogin.jsp
JavaMail SMTPAuthentication.java EnviaMail.java	RecuperaLogin.jsp ConsultaRegistro.jsp

Figura 3.1. Clases, librerías y JSP's del sistema.

Java DataBase Connectivity (JDBC) es una tecnología Java que permite a las aplicaciones interactuar directamente con motores de bases de datos relacionales. Esta API es independiente del sistema operativo donde se ejecuta y de la Base de datos a la que se accede.

JDBC ofrece el paquete `java.sql` con el que es posible trabajar con bases de datos. Además de un driver que utiliza la JVM para traducir las invocaciones JDBC en invocaciones que la base de datos entiende.

La clase `ManejaBD.java` establece la conexión con la base de datos mediante la API de JDBC y define los parámetros generales de dicha conexión, de tal forma que la clase se aplica para cualquier base de datos en MySQL. En la figura 3.2 se explica el código de esta clase.


```
//Paquete de la clase en el que se maneja la base en la aplicación
package unam.bd;

//Se importa el paquete que proporciona la API de JDBC
import java.sql.*;

//Otras librerías útiles
import java.util.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
public class ManejaBD {

    //Variable privada global de tipo Connection.
    private Connection con;

    //Constructor
    public ManejaBD() {}

    //Método general que inicia la conexión con los parámetros correspondientes
    public void estableceConexion(String driver, String url, String login, String
    pass) throws Exception {
        Class cl = Class.forName(driver);
        DriverManager.registerDriver((Driver)cl.newInstance());
        con = DriverManager.getConnection( url, login, pass);
    }

    //Metodo que cierra la conexion
    public void cerrar() throws Exception {
        if (con != null) {
            con.close();
        }
    }
}
```

Figura 3.2. Código de la clase ManejaBD

Además, se creó la clase ConsultaBD.java útil para hacer consultas y modificaciones a la base. El código se muestra en la figura 3.3.

```
//Paquete de la clase
package unam.bd;

//Libreria util para el manejo de los resultados de las consultas
import java.util.*;

public class ConsultaBD {

    ManejaBD mdb = new ManejaBD();
    private Vector vConsultaBD = new Vector();
    private String ctrlCnx = "", urlCnx = "", usr = "", passwd = "";

    //Constructor
    public ConsultaBD() {
    }

    //Se crea la conexión a la base de datos MySQL
    public void conexionMySQL() throws Exception {
        String driver = "com.mysql.jdbc.Driver";

        String url = "jdbc:mysql://localhost:3306/admin_usuarios";
        String usuario = "root";
        String password = "";
        mdb.establishConexion(driver, url, usuario, password);
    }

    //Metodo para hacer consultas a la base dada una cadena como sentencia
    public Vector consultaTabla(String cadSql) throws Exception {
        if (cadSql.equals("")) {
            return new Vector();
        }
        conexionMySQL();
        vConsultaBD = mdb.buscar(cadSql);
        cerrar();
        return vConsultaBD;
    }

    //Metodo que actualiza la base con una cadena recibida como sentencia
    public Vector modifica(String cadSql) throws Exception {
        if (cadSql.equals("")) {
            return new Vector();
        }
        conexionMySQL();
        mdb.modifica(cadSql);
        cerrar();
        return vConsultaBD;
    }
}
```

Figura 3.3. Código de la clase ConsultaBD

3.1 Autenticación de usuarios.

La acción de autenticar significa dar seguridad de que alguien o algo es lo que representa o parece. En este módulo se lleva a cabo el proceso de acreditación del usuario para establecer una sesión activa válida dentro del Sistema Integral de Administración y Control de Usuarios Online.



Figura 3.4. Pantalla inicial del Sistema Integral de Administración y Control de Usuarios Online

En la figura 3.4 se muestra la página login.jsp del Sistema que es donde el usuario debe ingresar su nombre de cuenta y su contraseña. Mediante código java y sus librerías correspondientes, se hace la conexión a la base de datos para verificar que los datos ingresados por el usuario existen y además son correctos.

Una vez verificado el acceso, se inicia la sesión del usuario dentro del Sistema para que pueda realizar sus actividades.

La consulta a la base se hace a través del método consultaTabla(String consulta) de la clase ConsultaBD con los datos del usuario. En la figura 3.5 se describe el código Java correspondiente.

```
// Se declaran variables útiles para obtener los datos
Vector vLogin = new Vector();
Hashtable htLogin = new Hashtable();
String sisIdAdmin = new String();
//Se hace la consulta a la base de datos
try{
    vLogin = consulta.consultaTabla(
        "SELECT * FROM USERS WHERE USER_NOMBRE_CUENTA =
        '"+usuario+"' AND USER_ESTATUS='V'"
    );
} catch(Exception ex){
    ex.printStackTrace();
}

//Si existe usuario valida contraseña
if(vLogin.size(>0){
    String clave = aes.regresaPwd(usuario);
    //Se compara la contraseña escrita con la almacenada en la base
    if(pwd.equals(clave)){
        //Si la contraseña es correcta valida rol
        String userId = (String)htLogin.get("USER_ID");
        Vector vRol = new Vector();
        try{
            vRol = consulta.consultaTabla(
                "SELECT A.SISTEMA_ID ,B.SISTEMA_DESCRIP_C FROM
                "+ "CONTROL_ACCESOS A, SISTEMAS B WHERE
                A.USER_ID"+ "="+userId+" AND A.TIPOROL_ID = 1 AND
                A.CTLAC_ESTATUS"+
                "='V' AND A.SISTEMA_ID=B.SISTEMA_ID"
            );
        } catch(Exception ex){
            ex.printStackTrace();
        }
        //Si el identificador del rol es de Administrador se manda a la pantalla
        //del menú que le corresponde.
        if(vRol.size()==1){
            Hashtable htRol=(Hashtable)vRol.get(0);
            sisIdAdmin=(String)htRol.get("SISTEMA_ID");
            session.setAttribute("usuario",usuario);
            session.setAttribute("sisIdAdmin",sisIdAdmin);
            //Codigo html

            //Si existe el rol de administrador para más de un sistema,
            //se preguntara sobre cual se desea administrar
        }else if(vRol.size(>1){
            //Codigo html
        }
    }
}
```

```
//Si el usuario no existe con el rol de administrador se envía a la
//pagina con menú de usuario normal.
} else if(vRol.size()==0){
    session.setAttribute("usuario", usuario);

//Si la contraseña es incorrecta
}else
    msgLogin = "Proceso de login fallido";

//Si no existe el usuario
}else
    msgLogin = "Proceso de login fallido";
```

Figura 3.5. Código de la página login.jsp que autentica al usuario.

3.2 Generación de responsiva PDF

Dentro del Sistema Integral de Administración y Control de Usuarios Online se generan algunos archivos en formato de documento portátil (PDF) generados a partir de la librería iText de Java que permite crear, manipular y exportar documentos PDF de forma dinámica. Los datos se escriben en un archivo o desde un servlet que es un programa que se ejecuta en el servidor actuando como una capa intermedia entre el navegador y las aplicaciones del servidor para proveer páginas web dinámicas y personalizadas, utilizando para este objetivo el acceso a bases de datos, flujos de trabajo y otros recursos.

La clase ResponsivaPDF (figura 3.6) genera un documento, en específico, el formato de una responsiva en PDF, con los datos del usuario mediante iText. En la figura 3.7 se muestra dicho documento ya generado, de tal forma que el usuario pueda imprimirlo, enviarlo o almacenarlo según requiera.

```
//El paquete de la clase
package unam.util;

//Librerías de la API iText para generacion de PDF
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.*;

//Se define el tipo de salida que genera el servlet
response.setContentType( "application/pdf" );

//Se define la variable que mostrara la salida
OutputStream out=response.getOutputStream();
```

```
//Definimos un nuevo documento para escribir en el los datos del PDF
Document documento = new Document(PageSize.LETTER,40,40,90,50);

//Variable que escribe el documento mediante la salida generada.
PdfWriter writer = PdfWriter.getInstance(documento,out);

//Fuentes
Font ffolio = new Font(Font.getFamily("ARIAL"), 10,
Font.BOLD,BaseColor.RED);
Font fetiquetas = new Font(Font.getFamily("ARIAL"), 10, Font.BOLD);
Font fdatos = new Font(Font.getFamily("ARIAL"), 10, Font.NORMAL);

//Parrafos
Paragraph folio;
folio = new Paragraph("Folio: 01"+userId,ffolio);
folio.setAlignment(Element.ALIGN_RIGHT);
folio.setSpacingAfter(10);

Paragraph datosUsuario;
datosUsuario = new Paragraph("DATOS DEL USUARIO",fetiquetas);
datosUsuario.setAlignment(Element.ALIGN_CENTER);
datosUsuario.setSpacingAfter(5);

Paragraph datosSistema;
datosSistema = new Paragraph("DATOS DEL SISTEMA",fetiquetas);
datosSistema.setAlignment(Element.ALIGN_CENTER);
datosSistema.setSpacingBefore(10);
datosSistema.setSpacingAfter(5);

Paragraph manifiesto;
String mani;
mani = "MANIFIESTO A USTED MI RESPONSABILIDAD EN EL USO DE
LA CUENTA CON EL NOMBRE DE USUARIO " +usuario+" DENTRO DEL
SISTEMA DE "+sistema+", CONFORME A LAS SIGUIENTES
DISPOSICIONES:";
manifiesto = new Paragraph(mani,fdatos);
manifiesto.setAlignment(Element.ALIGN_JUSTIFIED);
manifiesto.setSpacingBefore(20);
manifiesto.setSpacingAfter(10);
manifiesto.setIndentationLeft(55);
manifiesto.setIndentationRight(55);



//Tablas
float[] columnas = {20.0f,80.0f};
PdfPTable tUsuario;
tUsuario = new PdfPTable(2);
```

```
tUsuario.setWidths(columnas);
//Se define la celda predeterminada para la tabla de datos del usuario
//Alineación
tUsuario.getDefaultCell().setHorizontalAlignment(Element.ALIGN_LEFT);
tUsuario.getDefaultCell().setVerticalAlignment(Element.ALIGN_MIDDLE);
//Altura
tUsuario.getDefaultCell().setFixedHeight(20);

//Se agregan los datos a la tabla de datos del usuario
tUsuario.addCell(new Paragraph("Nombre completo",fetiquetas));
tUsuario.addCell(new Paragraph(nombreCompleto,fdatos));
tUsuario.addCell(new Paragraph("CURP",fetiquetas));
tUsuario.addCell(new Paragraph(curp,fdatos));
tUsuario.addCell(new Paragraph("RFC",fetiquetas));
tUsuario.addCell(new Paragraph(rfc,fdatos));
tUsuario.addCell(new Paragraph("Correo electrónico",fetiquetas));
tUsuario.addCell(new Paragraph(email,fdatos));
tUsuario.addCell(new Paragraph("Facultad",fetiquetas));
tUsuario.addCell(new Paragraph(facultad,fdatos));

//Se agrega la tabla al documento
documento.open();
documento.add(folio);
documento.add(datosUsuario);
documento.add(tUsuario);
documento.add(datosSistema);
documento.add(tSistema);
documento.add(manifiesto);
documento.close();
```

Figura 3.6. Código de la clase ResponsivaPDF.java

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE INGENIERÍA	
FCU-01	Responsiva de cuenta de usuario	Fecha: 28/Agosto/2012

Folio: 0115

DATOS DEL USUARIO

Nombre completo	IOVANI JUAREZ GARCIA
CURP	JUGI870629HDFRRV00
RFC	JUGI870629E21
Correo electrónico	ijuarezg@mail.issfam.gob.mx
Facultad	FACULTAD DE INGENIERIA

DATOS DEL SISTEMA

Usuario	ijuarezg
Sistema	ADMINISTRACION Y CONTROL DE USUARIOS
Rol	ADMINISTRADOR

MANIFIESTO A USTED MI RESPONSABILIDAD EN EL USO DE LA CUENTA CON EL NOMBRE DE USUARIO ijuarezg DENTRO DEL SISTEMA DE ADMINISTRACION Y CONTROL DE USUARIOS, CONFORME A LAS SIGUIENTES DISPOSICIONES:

1. La cuenta de usuario y la contraseña son de carácter estrictamente confidencial y queda bajo mi responsabilidad la custodia de las mismas así como su uso adecuado dentro del sistema y la información que en éste se maneja.
2. Me obligo a cambiar mi contraseña al inicio de la primera sesión
3. Reconozco y acepto que cualquier uso inadecuado que se detecte será objeto de las sanciones correspondientes conforme a lo dispuesto en la legislación aplicable.
4. Al término de mi responsabilidad dentro de la institución, solicitaré por escrito la cancelación de esta cuenta de usuario que se me está asignando.

Acepto las disposiciones arriba señaladas.
IOVANI JUAREZ GARCIA
1847

NOTA: FAVOR DE AGREGAR A LA RESPONSIVA UNA COPIA DE SU IDENTIFICACIÓN OFICIAL CON FIRMA PARA SU EXPEDIENTE.

1

Figura 3.7. Responsiva en formato PDF generada mediante código Java.

3.3 Registro de usuarios

A través de este módulo se registra la información de un usuario para crear su cuenta en el Sistema Integral de Administración y Control de Usuarios Online.

The screenshot shows a web form titled "SISTEMA INTEGRAL DE ADMINISTRACIÓN Y CONTROL DE USUARIOS ONLINE". It is divided into two main sections: "DATOS DEL USUARIO" and "DATOS DEL LOGIN".

DATOS DEL USUARIO:

- *Número de cuenta: [input field]
- *Facultad o Escuela: SELECCIONE... [dropdown menu]
- *Nombre(s): [input field]
- *Apellido paterno: [input field]
- *Apellido materno: [input field]
- CURP: [input field]
- RFC: [input field]
- *Correo electrónico: [input field]

DATOS DEL LOGIN:

- *Usuario: [input field]
- *Sistema: SELECCIONE... [dropdown menu]
- *Rol: SELECCIONE... [dropdown menu]

Below the form, there is a captcha instruction: "Ingrese los caracteres que ve en la imagen, sin espacios." The captcha image shows the characters "y k d f k b" in a blue box. At the bottom of the form are two buttons: "Enviar" and "Regresar".

Figura 3.8. Vista de la página SolicitaUsuario.jsp

En la página SolicitaUsuario.jsp (figura 3.8) se ingresan los datos solicitados y al terminar el registro se envía la información para que sea insertada en la base de datos.

Antes de enviarse el formulario, se validan los campos tratando de evitar datos incorrectos en la base de datos así como posible spam.

Estas validaciones incluyen tanto formato de los campos como datos repetidos, además de campos vacíos u obligatorios.

También se incluye un captcha (Prueba de Turing Pública y Automática para Diferenciar Máquinas y Humanos) para evitar a los robots de spam.

A través del método modifica () de la clase ConsultaBD se inserta el registro en la base de datos. El código correspondiente se describe en la figura 3.9.

```
//Primero a la tabla USERS que contiene la información del usuario:
vDatos = consulta.modifica(
    "INSERT INTO USERS ( USER_NOMBRE, USER_APE_PAT,
    USER_APE_MAT, USER_NUM_CUENTA, USER_RFC,
    USER_CURP, USER_EMAIL, USER_NOMBRE_CUENTA,
    USER_PASSWORD,FAC_CVE,USER_ESTATUS)

    VALUES(nombre, apePat, apeMat, numCta, rfc, curp, email, usuario,
    AES_ENCRYPT(clave, usuario), facCve ,V )"
);

//Después a la tabla CONTROL_ACCESOS para que tenga acceso al sistema:
vDatos = consulta.modifica(
    "INSERT INTO CONTROL_ACCESOS (USER_ID, SISTEMA_ID,
    TIPOROL_ID, CTLAC_FECHORA_A, CTLAC_ESTATUS) VALUES
    (userId, sisId, rolId, now(), C)"
);
```

Figura 3.9. Código que actualiza la base de datos al registrarse un nuevo usuario.

3.4 Recuperación de login

El módulo de recuperación de login ofrece al usuario, ya registrado, la posibilidad de recuperar sus datos de acceso en caso de que así lo requiera. Esto es, si ha olvidado su nombre de usuario o contraseña.

En la página RecuperaLogin.jsp (Figura 3.10) se debe ingresar el nombre de usuario (nombre de la cuenta) o el correo electrónico con el que se dio de alta.

Si el sistema encuentra el registro correspondiente, envía el correspondiente correo electrónico con los datos de acceso del usuario, como se muestra en la figura 3.11.



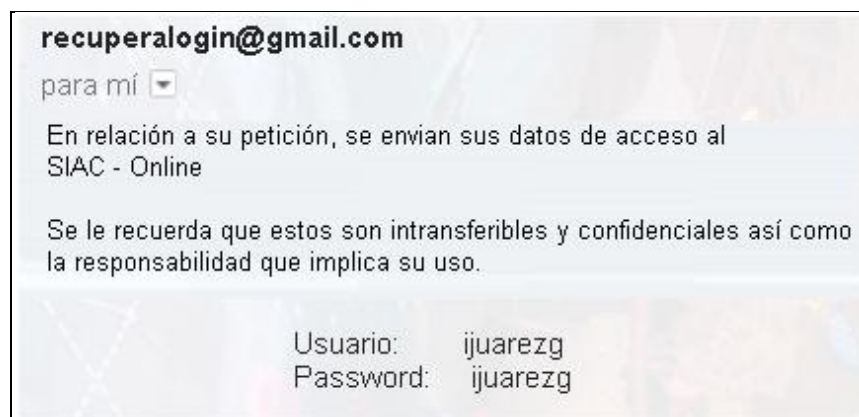
SISTEMA INTEGRAL DE ADMINISTRACIÓN Y CONTROL DE USUARIOS ONLINE

Para recuperar su contraseña ingrese su nombre de usuario:

o el correo electrónico que registró

correo electrónico:

Figura 3.10. Pantalla RecuperaLogin.jsp



recuperalogin@gmail.com
para mí ▾

En relación a su petición, se envían sus datos de acceso al SIAC - Online

Se le recuerda que estos son intransferibles y confidenciales así como la responsabilidad que implica su uso.

Usuario: ijuarezg
Password: ijuarezg

Figura 3.11. Correo electrónico con datos de acceso

Para el envío del correo electrónico se utiliza la API de JavaMail a través de dos clases:

SMTPAuthentication

Proporciona la cuenta y la contraseña que enviarán el correo a nombre del Sistema Integral de Administración y Control de Usuarios Online. En la figura 3.12 se explica el código correspondiente.

```
//Importacion de librería de JavaMail
import javax.mail.PasswordAuthentication;

public class SMTPAuthentication extends javax.mail.Authenticator{

    public PasswordAuthentication getPasswordAuthentication(){

        //Cuenta de correo electrónico asociada al Sistema
        String username = "recuperalogin@gmail.com";
        //Contraseña de la clave de la cuenta de correo electrónico
        String password = "sistemaglobal";

        //Devuelve un login valido
        return new PasswordAuthentication(username, password);
    }
}
```

Figura 3.12. Código de la clase SMTPAuthentication.java

EnviaMail.java

Se establece la comunicación con el servidor de correo electrónico mediante el protocolo SMTP y se hace el envío correspondiente de usuario y contraseña. Como se muestra en la figura 3.13.

```
import java.util.Properties;

//Librerias de JavaMail
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class EnviaMail {
    //Constructor
    public EnviaMail() {}
}
```


3.5 Menú de usuarios

Una vez que el usuario se ha autenticado dentro del Sistema, es dirigido a la página correspondiente dependiendo del rol que tenga activo.

Existen en primera instancia dos menús, uno para administrador y el segundo para usuarios “normales”.

Si el usuario es un administrador es dirigido a la página AdminLogin.jsp que tiene el menú correspondiente, como se muestra en la figura 3.14.



Figura 3.14. Menu de administrador

El usuario sin privilegios de administrador es dirigido a la página UserLogin.jsp con su menú correspondiente (figura 3.15).



Figura 3.15. Menu de usuario

Administración de usuarios

Como se ve en las figuras 3.14 y 3.15, el menú de administrador tiene una opción adicional que le permite habilitar y cancelar usuarios, de acuerdo a la normatividad y el proceso correspondiente.

En esta sección se muestra la lista de todos los usuarios registrados bajo el mismo sistema que el administrador, como se ve en la figura 3.16.

Folio: <input type="text"/> <input type="button" value="Buscar"/>				
Usuarios registrados: 14				
NOMBRE COMPLETO	FACULTAD	NUM. DE CUENTA	USUARIO	FOLIO
1. IOVANI JUAREZ GARCIA	FACULTAD DE INGENIERIA	303227722	ijuarezg	15
2. SILVESTRE LOPEZ ALVAREZ	FACULTAD DE QUIMICA	303227723	chachito	16
3. ROSA MARIA GARCIA ALARCON	FACULTAD DE ARQUITECTURA	A100XXX	rmgarciaa	26
4. JOSE LUIS JUAREZ SANCHEZ	FACULTAD DE CIENCIAS	B300ZZZ	jjuarezs	81
5. ROBERTO ESQUIVEL SOTO	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	964	resquivels	83
6. JAIME LOPEZ NORBERTO	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	11	jlopezn	108
7. YOLANDA ROSAS CERVANTES	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	32	yrosasc	113
8. ARCELIA GRANADOS SANTANA	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	22	agranadoss	114
9. BENITO OLIVERA AGUILAR	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	23	bolveraa	115
10. IRMA ALICIA GUZMAN PEREZ	FACULTAD DE CIENCIAS POLITICAS Y SOCIALES	236	iaguzmanp	118

Figura 3.16 Interfaz de administrador - Usuarios registrados

Dentro de esta opción, el administrador puede buscar a un usuario ya sea por folio, que se imprime en la responsiva, o por lista, es decir, de uno en uno seleccionando el renglón.

Al seleccionar un usuario, se muestra su información, así como los sistemas que tiene registrados (figura 3.17).

Folio: Buscar

Folio: 15
 Nombre: IOVANI JUAREZ GARCIA
 CURP: JUGI870629HDFRRV00
 RFC: JUGI870629E21
 Email: ing.juarez.unam@gmail.com

Facultad: FACULTAD DE INGENIERIA
 Número de cuenta: 303227722
 Usuario: ijuarezg
 Fecha de alta: 2011-12-13 00:00:00.0

SISTEMAS DEL USUARIO		
SISTEMA	ROL	ESTATUS
SIAC	ADMINISTRADOR	VIGENTE <input type="button" value="Cancelar"/>
REINSCRIPCION SEMESTRAL	ADMINISTRADOR	CANCELADO <input type="button" value="Vigente"/>
RIU	ADMINISTRADOR	CANCELADO <input type="button" value="Vigente"/>
SIAC	SUPERVISOR	CANCELADO <input type="button" value="Vigente"/>

Figura 3.17. Interfaz de administrador – Administración de sistemas del usuario

Se puede habilitar o cancelar un rol dentro de un sistema de acuerdo a los requerimientos de cada usuario, tomando en cuenta que el usuario solo debe tener un rol habilitado para cada sistema que tenga registrado.

Inicio

La opción inicio (figura 3.18) muestra la información del usuario que está ingresando al sistema.


USUARIO REGISTRADO:

Nombre: IOVANI JUAREZ GARCIA
 Facultad: FACULTAD DE INGENIERIA
 Número de cuenta: 303227722
 Email: ijuarezg@mail.com

Usuario: ijuarezg
 Estatus: VIGENTE

Figura 3.18 Información del usuario en la opción Inicio del menú

Dentro de esta opción, se ofrece además la posibilidad de actualizar algunos datos del mismo usuario, como se ve en la figura 3.19.



Nombre:
IOVANI

Apellido paterno:
JUAREZ

Apellido materno:
GARCIA

Facultad o Escuela:
FACULTAD DE INGENIERIA

CURP:
JUGI870629HDFRRV00

RFC:
JUGI870629E21

Correo electrónico:
ijuarezg@mail.com

Actualizar Regresar

Figura 3.19. Formulario de actualización de datos

El formulario de la figura 3.19 valida que la CURP y el RFC tengan el formato correcto y que el correo electrónico actualizado no esté ya registrado.

Sistemas

Esta sección del menú depende de las aplicaciones y/o sistemas que existen dentro de cada institución pues aquí es donde se hace la vinculación con cada uno de estos.

Existen dos opciones, la primera, Mis sistemas, muestra una lista que corresponde a los sistemas en que el usuario está dado de alta con un rol vigente. Figura 3.20.

A partir de esta lista es posible acceder a los diferentes sistemas que tendrán liga directa con el Sistema Integral de Administración y Control de Usuarios Online a través de identificadores únicos para el usuario, el sistema y el rol; el sistema ligado recibirá estos parámetros y los traducirá en información, privilegios, menús, etc.



Figura 3.20 Sistemas habilitados del usuario

La segunda opción, Solicita sistema, mostrada en la figura 3.21, despliega todos los sistemas que estén disponibles para que el usuario pueda solicitar, según requiera, un rol dentro de estos.

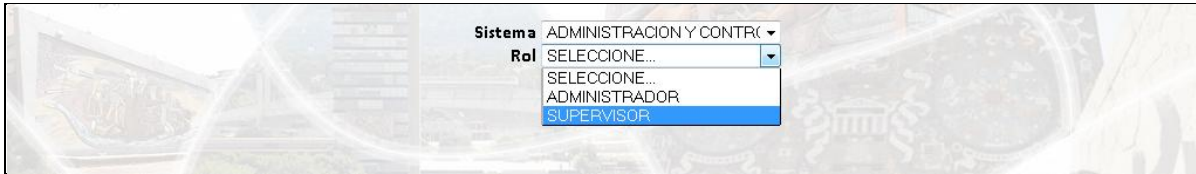


Figura 3.21 Pantalla de solicitud de rol dentro de un sistema.

Cada que el usuario solicite un nuevo rol en algún sistema, deberá imprimir su responsiva para firmarla y enviarla al administrador correspondiente para la aprobación de su solicitud.

Servicios

En esta parte del menú el usuario puede solicitar el cambio de su contraseña así como la impresión de alguna de sus responsivas.

En la primera opción, Cambiar contraseña, se debe ingresar la contraseña actual y la contraseña nueva, como se muestra en la figura 3.22, debiendo confirmar ésta ingresándola nuevamente para evitar errores o cambios no deseados.

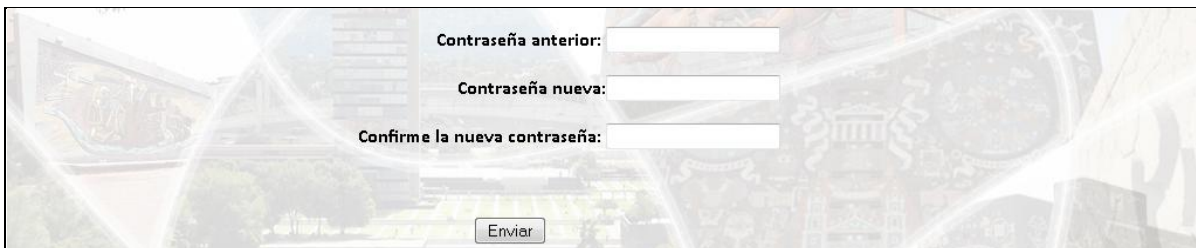


Figura 3.22. Formulario de cambio de contraseña.

El sistema hará, en primer lugar, una comparación con la contraseña actual para autenticar al usuario, después validará que la contraseña nueva y su confirmación coincidan y tengan mínimo 6 caracteres.

Si todo es correcto, la contraseña será actualizada y el sistema notificara al usuario de dicha acción. En caso contrario, el sistema también notificara del error, ya sea que la contraseña actual no sea correcta, que la contraseña nueva no tenga 6 o más caracteres o que la confirmación no coincida.

En la segunda opción (figura 3.23), el usuario puede ver las respuestas que se registran cada vez que solicita algún rol en cualquiera de los sistemas disponibles, esto con la finalidad de que pueda imprimir la que requiera para reenviarla, imprimirla, almacenarla, etc.

SISTEMA	ROL	
ADMINISTRACION Y CONTROL DE USUARIOS	ADMINISTRADOR	Imprimir responsiva
ADMINISTRACION Y CONTROL DE USUARIOS	SUPERVISOR	Imprimir responsiva
REINSCRIPCION SEMESTRAL	ADMINISTRADOR	Imprimir responsiva
REINSCRIPCION SEMESTRAL	OPERADOR	Imprimir responsiva
RIU	ADMINISTRADOR	Imprimir responsiva
RIU	OPERADOR	Imprimir responsiva

Figura 3.23. Tabla de respuestas generadas por un usuario.

Salir

Finalmente, la opción Salir, termina la sesión del usuario y lo regresa a la pantalla inicial del Sistema Integral de Administración y Control de Usuarios Online. En la figura 3.24 se muestra el código correspondiente.

```
//Se verifica si se ha seleccionado la opción Salir
if(!salida.equals("")){
    //Se termina la sesión
    session.invalidate();
    //Se redirige a la página de login.
    out.println("<script> location.href='../index.html' </script>");
}
```

Figura 3.24 Código de cierre de sesión.

CAPÍTULO 4
IMPLEMENTACIÓN, PRUEBAS Y RESULTADOS

A lo largo de este capítulo se realizará una descripción técnica de la instalación del Sistema Integral de Administración y Control de Usuarios Online así como de sus recursos. Además se detallarán las pruebas realizadas y los resultados obtenidos.

4.1 Base de datos

Una parte esencial del Sistema es la base de datos. Ésta es implementada mediante el entorno MySQL que es muy ligero, fácilmente administrable y sobre todo, rápido en la lectura de datos que en las aplicaciones web puede ser bastante intensiva.

En la figura 4.1 se describe el código SQL que se utiliza para crear la base, así como las tablas y los datos iniciales necesarios para el sistema.

```
-- Se crea la base de datos
CREATE DATABASE ADMON_USUARIOS;

-- Se crean las tablas
-- Tabla CONTROL_ACCESOS
CREATE TABLE IF NOT EXISTS `control_accesos` (
  `USER_ID` int(5) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DE USUARIO REGISTRADO',
  `SISTEMA_ID` int(4) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DE SISTEMA VIGENTE',
  `TIPOROL_ID` int(2) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DE TIPO DE ROL
ASIGNADO A LOS DIFERENTES SISTEMAS',
  `CTLAC_FECHORA_A` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE ALTA DEL
CONTROL DE ACCESO DEL USUARIO',
  `CTLAC_FECHORA_C` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE LA
MODIFICACION DE DATOS DE CONTROL DE ACCESO DEL USUARIO',
  `CTLAC_ESTATUS` varchar(1) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'SITUACION EN QUE SE ENCUENTRA EL CONTROL DE ACCESO DEL USUARIO (V: VIGENTE,
C: CANCELADO)',
  PRIMARY KEY (`USER_ID`,`SISTEMA_ID`,`TIPOROL_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- Tabla FACULTADES
CREATE TABLE IF NOT EXISTS `facultades` (
  `FACULTAD_CLAVE` int(2) NOT NULL DEFAULT '0',
  `FACULTAD_ABREV` varchar(5) DEFAULT NULL,
  `FACULTAD_DESC_C` varchar(25) DEFAULT NULL,
  `FACULTAD_DESC_L` varchar(100) DEFAULT NULL,
  `FACULTAD_REFERENCIA` varchar(30) DEFAULT NULL,
  `FACULTAD_FECHORA_A` datetime DEFAULT NULL,
  `FACULTAD_ESTATUS` varchar(1) DEFAULT NULL,
  PRIMARY KEY (`FACULTAD_CLAVE`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Tabla MENUS
CREATE TABLE IF NOT EXISTS `menus` (
  `MENU_ID` int(4) NOT NULL AUTO_INCREMENT COMMENT 'CLAVE IDENTIFICADOR DEL
MENU.',
  `MENU_NOMBRE_CORTO` varchar(50) COLLATE utf8_unicode_ci NOT NULL COMMENT
'NOMBRE CORTO DE MENU.',
  `MENU_DESCRIP` varchar(300) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
DESCRIPCION DETALLADA DEL MENU.',
```

```

`MENU_LIGA` varchar(250) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT 'DIRECCION
ELECTRONICA DE LA UBICACION DEL MENU PARA PODER ACCESAR',
`MENU_FECHORA_A` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE ALTA DEL
MENU.', `MENU_FECHORA_C` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE LA
MODIFICACION DE DATOS DEL MENU.',
`MENU_ESTATUS` varchar(1) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'SITUACION EN QUE SE ENCUENTRA EL MENU (V: VIGENTE, C: CANCELADO).',
PRIMARY KEY (`MENU_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

-- Tabla OPCIONES_MENU

```

CREATE TABLE IF NOT EXISTS `opciones_menu` (
`MENU_ID` int(4) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DEL MENU.',
`OPMEN_ID` int(4) NOT NULL AUTO_INCREMENT COMMENT 'CLAVE IDENTIFICADOR DE
LA OPCIÓN DEL MENU EXISTENTE',
`OPMEN_NOMBRE_CORTO` varchar(50) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT
NULL COMMENT 'NOMBRE CORTO DE LA OPCION DEL MENU',
`OPMEN_DESCRIP` varchar(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT
NULL COMMENT 'DESCRIPCION DE LA OPCION DEL MENU',
`OPMEN_LIGA` varchar(250) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL
COMMENT 'DIRECCION ELECTRONICA DE LA UBICACION DE LA OPCION DE MENU PARA
PODER ACCESAR',
`OPMEN_FECHORA_A` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE ALTA DEL
MENU.',
`OPMEN_ESTATUS` varchar(1) CHARACTER SET utf8 COLLATE utf8_unicode_ci DEFAULT NULL
COMMENT 'SITUACION EN QUE SE ENCUENTRA LA OPCION DEL MENU (V: VIGENTE, C:
CANCELADO).',
PRIMARY KEY (`MENU_ID`,`OPMEN_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

-- Tabla OPCIONES_SISTEMA

```

CREATE TABLE IF NOT EXISTS `opciones_sistema` (
`SISTEMA_ID` int(4) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DE SISTEMA
EXISTENTE',
`TIPOROL_ID` int(2) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DE TIPO DE ROL
ASIGNADO A LOS DIFERENTES SISTEMAS',
`MENU_ID` int(4) NOT NULL COMMENT 'CLAVE IDENTIFICADOR DEL MENU',
`OPDIS_FECHORA_A` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE ALTA DE LA
OPCION DEL SISTEMA',
`OPDIS_ESTATUS` varchar(1) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT 'SITUACION
EN QUE SE ENCUENTRA LAS OPCIONES DE SISTEMA (V: VIGENTE, C: CANCELADO)',
PRIMARY KEY (`SISTEMA_ID`,`TIPOROL_ID`,`MENU_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

-- Tabla SISTEMAS

```

CREATE TABLE IF NOT EXISTS `sistemas` (
`SISTEMA_ID` int(4) NOT NULL AUTO_INCREMENT COMMENT 'CLAVE IDENTIFICADOR DEL
SISTEMA',
`SISTEMA_DESCRIP_C` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'DESCRIPCION CORTA DEL SISTEMA REGISTRADO',
`SISTEMA_DESCRIP_L` varchar(500) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'DESCRIPCION LARGA DEL SISTEMA REGISTRADO',
`SISTEMA_LIGA` varchar(250) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'DIRECCION ELECTRONICA DE LA UBICACION DEL SISTEMA PARA PODER ACCESAR',
`SISTEMA_FECHORA_A` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DEL ALTA DE
SISTEMA',
`SISTEMA_ESTATUS` varchar(1) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
'SITUACION ACTUAL EN LA QUE SE ENCUENTRA LOS SISTEMAS EXISTENTES (V: VIGENTE,
C: CANCELADO)',
PRIMARY KEY (`SISTEMA_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

```

-- Tabla TIPOS_ROLES
CREATE TABLE IF NOT EXISTS `tipos_rol` (
  `TIPOROL_ID` int(2) NOT NULL AUTO_INCREMENT COMMENT 'CLAVE IDENTIFICADOR DE
  ROL',
  `TIPOROL_DESCRIP` varchar(30) CHARACTER SET utf8 COLLATE utf8_turkish_ci DEFAULT
  NULL COMMENT 'DESCRIPCION DEL ROL',
  `TIPOROL_FECHORA_A` datetime DEFAULT NULL COMMENT 'FECHA Y HORA DE ALTA DEL
  ROL',
  `TIPOROL_ESTATUS` varchar(1) COLLATE utf8_unicode_ci DEFAULT NULL COMMENT
  'SITUACION EN QUE SE ENCUENTRA EL ROL (V: VIGENTE, C: CANCELADO)',
  PRIMARY KEY (`TIPOROL_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- Tabla USERS
CREATE TABLE IF NOT EXISTS `users` (
  `USER_ID` int(5) NOT NULL AUTO_INCREMENT,
  `USER_NOMBRE` varchar(60) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_APE_PAT` varchar(50) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_APE_MAT` varchar(50) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_NUM_CUENTA` varchar(50) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_RFC` varchar(13) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_CURP` varchar(18) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_EMAIL` varchar(70) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_FECHORA_A` datetime DEFAULT NULL,
  `USER_FECHORA_C` datetime DEFAULT NULL,
  `USER_NOMBRE_CUENTA` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `USER_PASSWORD` varchar(1000) COLLATE utf8_unicode_ci DEFAULT NULL,
  `FAC_CVE` int(5) DEFAULT NULL,
  `USER_ESTATUS` varchar(1) COLLATE utf8_unicode_ci DEFAULT NULL,
  `USER_MOTIVO` varchar(50) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`USER_ID`),
  KEY `FAC_CVE` (`FAC_CVE`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

Figura 4.1 Código SQL que crea las tablas del Sistema Integral de Administración y Control de Usuarios Online

Después de que las tablas son creadas, se insertan los datos iniciales del Sistema Integral de Administración y Control de Usuarios Online. En la figura 4.2 se explica el código correspondiente.

```

-- Se inserta el usuario que será el administrador global (root)
INSERT INTO `users` (`USER_ID`, `USER_NOMBRE`, `USER_APE_PAT`, `USER_APE_MAT`,
`USER_NUM_CUENTA`, `USER_RFC`, `USER_CURP`, `USER_EMAIL`, `USER_FECHORA_A`,
`USER_NOMBRE_CUENTA`, `USER_PASSWORD`, `FAC_CVE`, `USER_ESTATUS`) VALUES
(1, 'IOVANI', 'JUAREZ', 'GARCIA', '303227722', 'JUGI870629E21', 'JUGI870629HDFRRV00',
'ijuarezg@gmail.com', now(), 'ijuarezg', 'ijuarezg', 13, 'V');

-- Además deben registrarse los permisos del administrador para que tenga acceso al sistema y pueda habilitar a
los -- nuevos usuarios. Para esto se inserta en la tabla CONTROL_ACCESOS la información necesaria.
INSERT INTO `control_accesos` (`USER_ID`, `SISTEMA_ID`, `TIPOROL_ID`, `CTLAC_FECHORA_A`,
`CTLAC_ESTATUS`) VALUES (1, 1, 1, now(), 'V');

```

```

-- Se registran los catálogos
-- Catalogo FACULTADES
INSERT INTO `facultades` (`FACULTAD_CLAVE`, `FACULTAD_DESC_C`, `FACULTAD_ABREV`,
`FACULTAD_DESC_L`, `FACULTAD_REFERENCIA`, `FACULTAD_ESTATUS`) VALUES
(1, 'FAC. ARQ.', 'FA', 'FACULTAD DE ARQUITECTURA', 'ARQUITECTURA', 'V'),
(2, 'FAC. CIENCIAS', 'FC', 'FACULTAD DE CIENCIAS', 'CIENCIAS', 'V'),
(3, 'FAC. C. POL. Y SOC.', 'FCPS', 'FACULTAD DE CIENCIAS POLITICAS Y SOCIALES', 'CIENCIAS
POLITICAS', 'V'),
(4, 'FAC. CONT. ADMON.', 'FCA', 'FACULTAD DE CONTADURIA Y ADMINISTRACION',
'CONTADURIA Y ADMON.', 'V'),
(5, 'FAC. DER.', 'FD', 'FACULTAD DE DERECHO', 'DERECHO', 'V'),
(6, 'FAC. ECON.', 'FE', 'FACULTAD DE ECONOMIA', 'ECONOMIA', 'V'),
(7, 'F.E.S. ACATLAN', 'FESAC', 'FACULTAD DE ESTUDIOS SUPERIORES ACATLAN', 'ACATLAN', 'V'),
(8, 'F.E.S. ARAGON', 'FESAR', 'FACULTAD DE ESTUDIOS SUPERIORES ARAGON', 'ARAGON', 'V'),
(9, 'F.E.S. CUAUTITLAN', 'FESC', 'FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN',
'CUAUTITLAN', 'V'),
(10, 'F.E.S. IZTACALA', 'FESI', 'FACULTAD DE ESTUDIOS SUPERIORES IZTACALA', 'IZTACALA', 'V'),
(11, 'F.E.S. ZARAGOZA', 'FESZ', 'FACULTAD DE ESTUDIOS SUPERIORES ZARAGOZA', 'ZARAGOZA',
'V'),
(12, 'FAC. FIL. Y LETRAS', 'FFL', 'FACULTAD DE FILOSOFIA Y LETRAS', 'FILOSOFIA', 'V'),
(13, 'FAC. ING.', 'FI', 'FACULTAD DE INGENIERIA', 'INGENIERIA', 'V'),
(14, 'FAC. MED.', 'FM', 'FACULTAD DE MEDICINA', 'MEDICINA', 'V'),
(15, 'FAC. MED. VET. Y ZOO.', 'FMVZ', 'FACULTAD DE MEDICINA VETERINARIA Y ZOOTECNIA',
'VETERINARIA', 'V'),
(16, 'FAC. ODONTO.', 'FO', 'FACULTAD DE ODONTOLOGIA', 'ODONTOLOGIA', 'V'),
(17, 'FAC. PSICO.', 'FP', 'FACULTAD DE PSICOLOGIA', 'PSICOLOGIA', 'V'),
(18, 'FAC. QUIM.', 'FQ', 'FACULTAD DE QUIMICA', 'QUIMICA', 'V'),
(19, 'ESC. NAC. DE A. PLASTICAS', 'ENAP', 'ESCUELA NACIONAL DE ARTES PLASTICAS', 'ARTES
PLASTICAS', 'V'),
(20, 'ESC. NAC. DE ENF. Y OBST.', 'ENEO', 'ESCUELA NACIONAL DE ENFERMERIA Y OBSTETRICIA',
'ENFERMERIA', 'V'),
(21, 'ESC. NAC. DE MUSICA', 'ENM', 'ESCUELA NACIONAL DE MUSICA', 'MUSICA', 'V'),
(22, 'ESC. NAC. DE T. SOCIAL', 'ENTS', 'ESCUELA NACIONAL DE TRABAJO SOCIAL', 'TRABAJO
SOCIAL', 'V');

-- Catalogo TIPOS_ROLES
INSERT INTO `tipos_rols` (`TIPOROL_ID`, `TIPOROL_DESCRIP`, `TIPOROL_FECHORA_A`,
`TIPOROL_ESTATUS`) VALUES
(1, 'ADMINISTRADOR', '2011-12-06 11:57:29', 'V'),
(2, 'SUPERVISOR', '2011-12-06 11:57:52', 'V'),
(3, 'OPERADOR', '2011-12-20 10:50:59', 'V'),
(4, 'ANALISTA', '2011-12-20 10:51:10', 'V'),
(5, 'CONSULTA', '2011-12-20 10:50:59', 'V');

-- Catalogo de SISTEMAS
-- Este catalogo dependerá de los servicios y/o aplicaciones que cada organización ofrezca, sin embargo, el primer
--- sistema que debe registrarse es el Sistema Integral de Administración y Control de Usuarios Online.
INSERT INTO `sistemas` (`SISTEMA_ID`, `SISTEMA_DESCRIP_C`, `SISTEMA_DESCRIP_L`,
`SISTEMA_LIGA`, `SISTEMA_FECHORA_A`, `SISTEMA_ESTATUS`) VALUES
(1, 'SISTEMA INTEGRAL DE ADMINISTRACIÓN Y CONTROL DE USUARIOS ONLINE', 'SISTEMA DE
CONTROL DE ACCESOS', 'http://172.16.3.120:8989/AdmonUNAM/prueba.jsp', now(), 'V'),
(2, 'REINSCRIPCION SEMESTRAL', 'MODULO DE REINSCRIPCION DE ALUMNOS',
'http://172.16.3.120:8989/AdmonUNAM/prueba.jsp', now(), 'V'),
(3, 'RIU', 'RED INALAMBRICA UNIVERSITARIA', 'http://172.16.3.120:8989/AdmonUNAM/prueba.jsp',
now(), 'V');

```

Figura 4.2 Código SQL para insertar los datos iniciales del Sistema Integral de Administración y Control de Usuarios Online.

Una vez que los datos iniciales y los catálogos se han cargado en la base de datos, puede habilitarse el Sistema Integral de Administración y Control de Usuarios Online para que los nuevos usuarios se registren y envíen las respuestas correspondientes para que el administrador global habilite sus cuentas y puedan acceder al Sistema.

4.2 Servidor de aplicaciones

Para implementar el Sistema Integral de Administración y Control de Usuarios Online se utiliza el servidor de aplicaciones que ofrece Apache. En la figura 4.3 se describe la información del servidor.

Información de Servidor					
Versión de Tomcat	Versión JVM	Vendedor JVM	Nombre de SO	Versión de SO	Arquitectura de SO
Apache Tomcat/6.0.35	1.6.0_24-b24	Sun Microsystems Inc.	Linux	3.2.0-23-generic-pae	i386

Figura 4.3 Información del servidor de aplicaciones

4.3 Pruebas

La fase de pruebas se refiere al conjunto de actividades dentro del desarrollo de software que son implementadas durante todo este proceso o en cualquier momento del mismo. Las pruebas de software involucran las operaciones del sistema bajo condiciones controladas y evaluando los resultados.

Existen dos vertientes fundamentales de pruebas de software:

- Caja negra: se lleva a cabo sin tener conocimiento de la estructura o funcionamiento interno del sistema. Solo se conocen las entradas apropiadas que debe recibir la aplicación, así como las correspondientes salidas.
- Caja blanca: se necesita conocimiento específico del código para analizar los resultados.

Las pruebas realizadas al SIAC – Online están basadas en el principio de caja negra, sin embargo, también se realizaron otros tipos de pruebas como:

- Unitarias
- Integración
- Integridad de datos
- Seguridad y control de acceso
- Interfaz grafica de usuario (GUI)

En las siguientes páginas se explican únicamente las pruebas de caja negra para verificar que el SIAC-Online contempla y valida todos sus flujos alternos.

Registro de usuarios

Cuando un nuevo usuario solicita una cuenta dentro del sistema, es necesario que registre sus datos completamente. Aunque existen campos que no son indispensables para registrar al usuario, hay otros que, por el contrario, son obligatorios para crear la nueva cuenta. Estos datos se identifican mediante un asterisco (*) que los precede, sin embargo es necesario llevar a cabo una validación para evitar que estos campos queden vacíos.

En la figura 4.4 se muestran los datos obligatorios que el usuario ha dejado vacíos mediante un cambio de color de fuente.

SISTEMA INTEGRAL DE ADMINISTRACIÓN Y CONTROL DE USUARIOS ONLINE

DATOS DEL USUARIO

- *Número de cuenta:
- *Facultad o Escuela: SELECCIONE...
- *Nombre(s):
- *Apellido paterno:
- *Apellido materno:
- CURP:
- RFC:
- *Correo electrónico:

DATOS DEL LOGIN

- *Usuario:
- *Sistema: SELECCIONE...
- *Rol: SELECCIONE...

Ingrese los caracteres que ve en la imagen, sin espacios.

c o b c s k k

Enviar Regresar

Figura 4.4 Los campos marcados en rojo indican que son obligatorios para el registro de un usuario nuevo.

Cuando se han completado los campos para el registro, el sistema hace una validación adicional para verificar que los datos se han ingresado correctamente y además verifica que los campos de número de cuenta, correo electrónico y nombre de usuario no existan ya en la base de datos pues sirven como identificadores de cada cuenta de usuario.

En la figura 4.5 se aprecia la validación de la CURP, el RFC, el correo electrónico y la verificación de que el número de cuenta, el mismo correo y el nombre de usuario no están registrados en la base.



SISTEMA INTEGRAL DE ADMINISTRACIÓN Y CONTROL DE USUARIOS ONLINE

DATOS DEL USUARIO

- *Número de cuenta: 303227722
Este número ya está registrado
- *Facultad o Escuela: FACULTAD DE INGENIERIA
- *Nombre(s): IOVANI
- *Apellido paterno: JUAREZ
- *Apellido materno: GARCIA
CURP: JUGI
CURP incorrecto, verifique por favor.
RFC: RFC
RFC incorrecto, verifique por favor.
- *Correo electrónico: ing.juarez.unam@gmail.com
Este correo ya está registrado

DATOS DEL LOGIN

- *Usuario: ijuarezg
Este nombre de usuario ya está registrado
- *Sistema: REINSCRIPCION SEMESTRA
- *Rol: ADMINISTRADOR

Ingrese los caracteres que ve en la imagen, sin espacios.

6 3 w j p c
63wjpc

Figura 4.5 Validación de campos para evitar errores y duplicidad de datos en la base.

En la última parte del registro se pide un código captcha para evitar que los robots de spam registren usuarios en el sistema. También este campo tiene su respectiva validación, como se observa en la figura 4.6.



Ingrese los caracteres que ve en la imagen, sin espacios.

h s 6 2 j 4
676344

Intente de nuevo porfavor.

Enviar Regresar

Figura 4.6 Validación de captcha.

Inicio de sesión

En el modulo de inicio de sesión se ha implementado una validación que permite hasta 3 intentos fallidos de inicio de sesión con el fin de evitar ataques por fuerza bruta. En la figura 4.7 se ilustra el mensaje correspondiente que muestra el sistema.



Figura 4.7 Mensaje de error de inicio de sesión y conteo regresivo de intentos.

Cuando se han agotado los 3 intentos, el sistema bloquea los campos de usuario y contraseña, así como el botón que envía los datos (Figura 4.8). Estos se vuelven a activar reiniciando el explorador con el que se navega en internet.

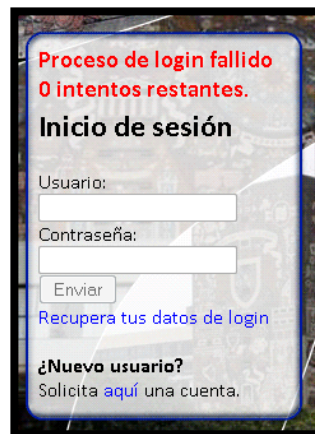


Figura 4.8 El inicio de sesión se bloquea después de 3 intentos fallidos.

Actualización de datos de usuario

Una vez que el usuario ha iniciado sesión, dentro del menú inicio a través de la opción Actualizar perfil es posible cambiar algunos datos del mismo usuario, es aquí donde se implementan de nuevo las validaciones correspondientes para la CURP, el RFC y el correo electrónico (Figura 4.9).



The screenshot shows a user profile update form with the following fields and values:

- Nombre:** IOVANI
- Apellido paterno:** JUAREZ
- Apellido materno:** GARCIA
- Facultad o Escuela:** FACULTAD DE INGENIERIA (dropdown menu)
- CURP:** JUGI870629H
- RFC:** JUGI8706
- Correo electrónico:** ing.juarez.unam@gmail.com

Validation messages are displayed in red text:

- CURP incorrecto, verifique por favor.**
- RFC incorrecto, verifique por favor.**
- Este correo ya existe, verifique por favor.**

Buttons at the bottom: Actualizar, Regresar

Figura 4.9. Validación de datos al actualizar el perfil del usuario.

Solicitud de sistema y rol

En la opción Solicitud de sistema del menú Sistemas es posible solicitar un nuevo rol dentro de un sistema, ya sea este rol el primero que se solicita en el sistema o uno distinto. Sin embargo si ya se ha solicitado el mismo rol para el mismo sistema, se muestra un mensaje que le indica al usuario que su solicitud ya se ha registrado (Figura 4.10).



The screenshot shows a form for requesting a system and role with the following fields and values:

- Sistema:** SISTEMA GLOBAL DE ADMIN (dropdown menu)
- Rol:** SELECCIONE... (dropdown menu)
- Enviar:** Button

A red message is displayed below the form: **Ya está registrado en este sistema con el rol solicitado**

Figura 4.10. Solicitud de sistema y rol ya registrados y el mensaje que muestra el sistema.

Cambio de contraseña

Esta opción, que se encuentra en el menú de Servicios, permite cambiar la contraseña de un usuario pasando por algunas validaciones. La primera verifica que la nueva contraseña tenga mínimo 6 caracteres (Figura 4.11).



The screenshot shows a web form for changing a password. It contains three input fields: 'Contraseña anterior:' with five dots, 'Contraseña nueva:' with four dots, and 'Confirme la nueva contraseña:' with four dots. Below the fields is a red error message: 'La contraseña debe tener al menos 6 caracteres'. At the bottom is a blue 'Enviar' button.

Figura 4.11. Validación de contraseña con al menos 6 caracteres.

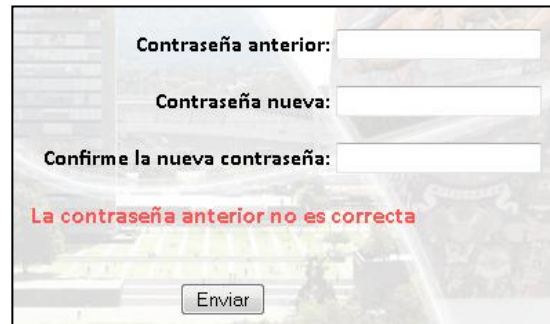
Además se verifica que no contenga caracteres especiales, en este caso son %, /, punto, coma, &, *, (,), [,], (,), {, }, -. Como se muestra en la figura 4.12, el sistema advertirá al usuario si ingresa alguno de los ya mencionados caracteres.



The screenshot shows a web form for changing a password. It contains one input field: 'Contraseña anterior:' with three dots. Below the field is a white error message box with the text: 'ERROR: No se aceptan caracteres especiales como %, /, punto, coma, &, *, (,), [,], (,), {, }, -'. At the bottom of the error box is a blue 'Aceptar' button. Below the error box is a blue 'Enviar' button.

Figura 4.12. Validación de contraseña sin caracteres especiales.

Posteriormente se valida que la contraseña anterior, es decir, la que se desea cambiar, sea correcta (Figura 4.13).



Contraseña anterior:

Contraseña nueva:

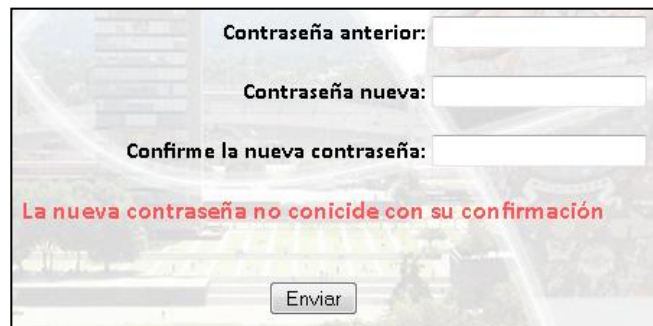
Confirme la nueva contraseña:

La contraseña anterior no es correcta

Enviar

Figura 4.13. Validación de contraseña anterior

Por último, se verifica que la contraseña nueva y su confirmación coincidan, de lo contrario el sistema informa al usuario como se ve en la figura 4.14 a continuación.



Contraseña anterior:

Contraseña nueva:

Confirme la nueva contraseña:

La nueva contraseña no concide con su confirmación

Enviar

Figura 4.14. Validación de contraseña nueva con su confirmación.

4.4 Resultados

Después de llevar a cabo las pruebas contempladas para el SIAC-Online, se corrigieron los errores o excepciones encontrados.

En cuanto a la operación, la interfaz del sistema es bastante intuitiva; los textos y ayudas que se proporcionan facilitan la navegación a través de ésta. El sistema resulta bastante amigable para nuevos usuarios y brinda las medidas básicas de seguridad necesarias para una aplicación que se encuentra en internet.

Los datos utilizados son reales de diferentes casos que se seleccionaron para cubrir la mayoría de módulos, validaciones y procesos del sistema.

El Sistema Integral de Administración y Control de Usuarios Online, entonces, se encuentra listo para cubrir el objetivo inicial tomando en cuenta además, ciclos de mantenimiento, actualización y mejora.

CAPÍTULO 5
CONCLUSIONES

El Sistema Integral de Administración y Control de Usuarios Online se ha desarrollado en lenguaje Java que es soportado por cualquier navegador web y que es un cliente ligero, independiente del sistema operativo y fácil de actualizar. No es necesario instalar software a miles de usuarios potenciales.

Lo anterior hace que el mismo sistema sea flexible y fácilmente adaptable a las necesidades de cada organización agregando o sustituyendo módulos según sea requerido.

La administración de usuarios, tanto internos como externos, de cualquier organización resulta de gran importancia sobre todo si en esta se ofrecen servicios.

Los principales objetivos de la administración de usuarios son:

- Autenticar la identidad de la persona que se conecta al sistema.
- Controlar el acceso a los recursos de la organización vía web.
- Auditar las acciones realizadas con la cuenta de usuario.

El uso de un login para cada usuario proporciona la autenticación de su identidad controlando así, mediante el rol que éste tenga dentro del sistema, los servicios a los que tendrá acceso.

Se lleva un registro, en la base de datos del sistema, de la fecha en que el usuario realice algún cambio en su sesión.

La implementación de un sistema que ayude a administrar usuarios y que además permita controlar los servicios que se ponen a disposición de éstos ha sido la principal motivación de este trabajo.

Su desarrollo ha permitido la detección de necesidades específicas en cuanto a la administración de usuarios se refiere.

El sistema resultante es una aplicación completa y adaptable al proceso de registro y administración de usuarios de cualquier organización.

Las lecciones aprendidas son aun más relevantes puesto que se ha llevado a cabo el proceso completo de desarrollo de software, llevando así los conceptos y teorías aprendidos en el salón de clase a la práctica real y exigente de la vida laboral y organizacional.

BIBLIOGRAFÍA Y REFERENCIAS

- Ingeniería del Software Un enfoque practico
Roger Pressman
McGraw Hill
6ta edición, 2005
- Sistemas de administración de bases de datos
Gerald V. Post
McGraw Hill
3ª edición, 2006
- Seguridad informática
Juan José Nombela
Editorial Paraninfo
1997
- Comparative Programming Languages
Wilson, Leslie B. (1993).
Addison-Wesley.
Second Edition.
- Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee
Naur, P. Y B. Randall (eds.),
NATO, 1969.
- Revista Digital Universitaria
10 de junio 2006
Volumen 7 Número 6
- Tecnologías de información y Herramientas para la Administración de proyectos de software
Omar Higinio Caballero Cervantes
Maestría en Administración de Tecnologías de Información, Instituto Tecnológico de Estudios Superiores de Monterrey, Campus Monterrey.
- Los poderes de la mente
Adam Smith
Editorial Pomaire
1979

-
- <http://maagtic.com/Documentos%20MAAGTIC/Forms/AllItems.aspx>
 - http://www.matem.unam.mx/~rajsbaum/cursos/web/presentacion_seguridad_1.pdf
 - http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software
 - <http://www.slideshare.net/javi2401/java-y-bases-de-datos-presentation>
 - http://es.wikipedia.org/wiki/Java_Database_Connectivity
 - <http://es.wikipedia.org/wiki/PDF>
 - <http://itextpdf.com/>
 - http://publib.boulder.ibm.com/infocenter/ltscnct/v2r0/index.jsp?topic=/com.ibm.lotus.connections.help/t_admin_common_config_mail_was.html
 - <http://lema.rae.es/drae/>
 - <http://www.slideshare.net/cliceduca/pruebas-de-software-2420588>
 - <http://informatica.iescuravalera.es/iflica/gtfinal/libro/x3697.html>
 - http://www.sui.gov.co/suibase/documentos/SUA_ManualAdministracionUsuarios.pdf
 - http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_Web
 - <http://www.enable-javascript.com/es/>
 - http://www.sat.gob.mx/sitio_internet/asistencia_contribuyente/preguntas_frecuentes/2_2294.html
 - <http://definicion.de/online/>
 - <http://www.informatica-hoy.com.ar/informatica-tecnologia-empresas/Ventajas-de-los-sistemas-online.php>
 - <http://www.losporques.com/tecnologia/que-son-los-sistemas-on-line.htm>
 - http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/viveros_s_ca/capitulo6.pdf
 - <http://ing-sw.blogspot.mx/2005/04/tipos-de-pruebas-de-software.html>