



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN**

**“SISTEMA DE ADMINISTRACIÓN Y CONTEO PARA  
PRESTADORES DE SERVICIO SOCIAL EN LA FES  
ARAGÓN (SICSS)”**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:

**INGENIERO EN COMPUTACIÓN**

**P R E S E N T A:**

**RICARDO BULMARO RIVERA RAMÍREZ**

**ASESOR DE TESIS:**

**M. EN C. MARCELO PÉREZ MEDEL**



**FES Aragón**

**MÉXICO, 2011.**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# AGRADECIMIENTOS

*A mi padre Bulmaro,*

*Por haberme dado la oportunidad y el privilegio de llegar hasta aquí, sin él simplemente hubiese sido imposible conseguirlo...Papá te dije que lo haría y ¡Aquí esta!, Gracias por todo.*

*A mis hermanos,*

*Ustedes que en todo momento me apoyaron y me han mostrado las diversas caras que tiene el destino, de no ser por ello difícilmente habría entendido la importancia de ser perseverante. César con tu capacidad para hacer de la vida un arte, Magali con tu gran carisma e inteligencia, me enorgullecen de ser mi sangre. ¡Los Adoro!*

*A mi familia,*

*Al ser un gran número de personas que la conforman no me alcanzan las palabras para darles gracias a todos y no quisiera excluir a ninguno aunque ustedes saben muy bien que los quiero mucho, a mis parientes García, Madrigal, Badillo, Cervantes, Castellón, Espejo, Escobar, Serrano, Carranza,, Rivera...por todo lo recibido y aprendido, ¡Muchas gracias!*

*A mi novia Ana Rosa*

*Gracias mi nena por apoyarme tanto en este laborioso trámite de titulación, y sobre todo, por llegar en el momento más apropiado para darme ese impulso del cual todos dependemos... el amor.*

*I Love You My rOsita...*

*A mi madre Luisa,*

*La más hermosa mujer que me ha dado la vida, amor y el pan de cada día... Gracias por seguir manteniendo ese gran poder de tener junta y feliz a la familia, ¡Te amo mamá!*

*A mi segunda madre,*

*La vida tuvo la gentileza de obsequiarme la compañía, protección y amor de una mujer a quien le debo muchísimo y a quien que hoy por hoy la quiero como a una madre, ¡Gracias por todo tía Flor!*

*A mi asesor y sinodales*

*Gracias a sus conocimientos y grandes trayectorias académicas me ha sido posible dar un paso más en mi vida profesional. Agradezco de verdad su tiempo empleado en este proyecto, así como sus sabias acreditaciones.*

# CONTENIDO

<b>INTRODUCCIÓN</b> .....	1
<b>1. Fundamentos de computación</b> .....	6
1.1 Antecedentes.....	6
1.2 Señal analógica y digital.....	7
1.3 Sistema binario.....	8
1.4 Generaciones de computadoras.....	9
1.4.1 Primera generación.....	9
1.4.2 Segunda generación.....	10
1.4.3 Tercera generación.....	11
1.4.4 Cuarta generación.....	12
1.4.5 Quinta generación.....	13
1.4.6 Sexta generación.....	14
1.5 Compilación e interpretación.....	15
1.6 Navegadores de internet.....	16
<b>2. Desarrollo Web</b> .....	18
2.1 Lenguajes de programación Web.....	19
2.1.1 HTML.....	19
2.1.2 JavaScript.....	19
2.1.3 PHP.....	20
2.1.4 CSS.....	21
2.1.5 XML.....	22
2.2 Base de datos PostgreSQL.....	22
2.3 AJAX.....	25
<b>3. Análisis del Sistema de Administración para Conteo de horas del Servicio Social (SICSS)</b> .....	27
3.1 UML y Casos de uso.....	29
3.2 Diagrama de casos de uso.....	50
<b>4. Diseño y construcción del Sistema de Administración para Conteo de horas del Servicio Social (SICSS)</b> .....	53
4.1 Marcos HTML (Frames).....	54
4.2 Animación Flash (Cabecera <i>top-frame</i> ).....	57
4.3 Menú de botones Flash ( <i>left-frame</i> ).....	58
4.4 Marco de contenido ( <i>main-frame</i> ).....	60
4.4.1 Sección de “Inicio”.....	60
4.4.2 Sección de “Administrar”.....	62
4.4.2.1 Menú de Administradores.....	62
4.4.2.2 Registros.....	66
4.4.2.3 Consultas.....	68
4.4.2.4 Modificación de datos.....	73

4.4.2.5	Agregar o quitar lugares y direcciones IP.....	77
4.4.2.6	Configuración del sistema de restricción de IP.....	79
4.4.3	Sección de “Iniciar de sesión”.....	80
4.4.3.1	Menú de usuarios.....	81
4.4.3.2	Registrar usuarios.....	82
4.4.3.2.1	Esquema de PostgreSQL.....	88
4.4.3.3	Autenticación de registro de usuarios.....	96
4.4.3.4	Iniciar conteo.....	98
4.4.3.5	Usuarios activos.....	100
4.4.3.6	Terminar conteo.....	101
4.4.3.7	Forzar baja.....	102
4.4.4	Sección de “Acerca de”.....	104
<b>CONCLUSIONES</b> .....		107
<b>REFERENCIAS</b> .....		108
<b>GLOSARIO</b> .....		110

## INDICE DE FIGURAS

Figura 3.1	- Diagrama de casos de uso del SICSS.....	52
Figura 4.1	- Marcos de la página web.....	56
Figura 4.2	- Cabecera de la página web.....	58
Figura 4.3	- Menú de botones animados de la página web.....	60
Figura 4.4	- Menú de botones animados de la página web.....	61
Figura 4.5	- Menú de administradores diseñado en HTML.....	64
Figura 4.6	- Menú de administradores editado con CSS y JavaScript.....	65
Figura 4.7	- Formulario de registro para asesores.....	66
Figura 4.8	- Formulario de registro para prestadores de servicio social.....	67
Figura 4.9	- Consulta de asesores registrados.....	69
Figura 4.10	- Consulta de usuarios registrados.....	70
Figura 4.11	- Consulta individual a datos del usuario a través del número de cuenta.....	70
Figura 4.12	- Consulta a la información de los usuarios que han iniciado su conteo.....	71
Figura 4.13	- Consulta a registros con información de cada inicio de conteo de tiempo.....	72
Figura 4.14	- Consulta individual de registros de cada inicio de conteo de tiempo.....	73
Figura 4.15	- Edición de registros de asesores.....	74
Figura 4.16	- Formulario para actualizar datos de asesores.....	74
Figura 4.17	- Borrado de un registro de asesor.....	75
Figura 4.18	- Consulta a registros de usuarios para editar su tiempo acumulado.....	76
Figura 4.19	- Pantalla consecutiva a edición de horas de usuarios.....	77
Figura 4.20	- Agregar o quitar direcciones IP y lugares de labor.....	78
Figura 4.21	- Sección para modificar estado de sistema de restricción de direcciones IP.....	79
Figura 4.22	- Pantalla de bienvenida al inicio de sesión del asesor.....	81
Figura 4.23	- Formulario de registro de usuarios.....	83
Figura 4.24	- Mensajes de alerta al dejar un campo de formulario en blanco.....	85
Figura 4.25	- Pantalla de bienvenida al usuario cuando inicia su conteo.....	98

Figura 4.26 - Consulta de usuarios activos realizada por asesores	99
Figura 4.27 - Pantalla de término de conteo del usuario	100
Figura 4.28 - Usuarios activos para forzar término de conteo exclusivamente por el asesor	101
Figura 4.29 - Pantalla consecuente al forzado de término de conteo de tiempo del usuario	102
Figura 4.30 - Sección para conocer más sobre el desarrollo de la página SICSS	103
Figura 4.31 - Pantalla con datos para contactarnos, ubicada en nube de etiquetas	104
Figura 4.32 - Sección de comentarios o firmas que se deseen dejar en la página	104

## INTRODUCCIÓN

---

---

En el mundo nuevo los tiempos están siendo alterados gravemente por el ser humano, ya que se han creado en su totalidad necesidades que no son naturales, es decir, recursos materiales, descubiertos y ahora utilizados como materia indispensable para nuestras vidas cotidianas.

Puede que sólo sea parte de una evolución, sin embargo es lamentable que la ambición sea un factor importante que los capitalistas utilicen para diseñar nuevas estrategias que nos hagan depender de estos grandiosos avances; no obstante sólo es la contraparte de todo un mundo de comodidad e innovación del cual todos salimos beneficiados.

No hace falta fundamentar la historia para deducir que la tecnología ha sido y es la base principal de la evidente evolución humana.

Pero ¿De qué manera nos apoyamos en la tecnología para evolucionar? Hablar de avances tecnológicos remontaría a épocas antiguas en donde sigue siendo clara la mejora a nuestro mundo, sin embargo únicamente se necesita citar el ámbito a tratar en este proyecto, la tecnología aplicada mediante sistemas de cómputo.

¿Por qué hemos dejado de realizar tareas iterativas por asignárselas a alguna máquina? Estos sistemas, en la actualidad, representan ya la parte fundamental administrativa de casi cualquier empresa; se han introducido rápidamente gracias a su gran velocidad para procesar información, así como controlarla y almacenarla indefinidamente, cumpliendo así el objetivo principal por el cual fueron creadas las computadoras: automatización y velocidad de procedimientos.

El óptimo desempeño de una computadora está en función de sus componentes, sin embargo también es posible desarrollar un sistema funcional en una computadora más austera.

Actualmente la UNAM y muchas otras instituciones dedicadas a prestar servicios, cuentan con opción de ingresar a prestadores de servicio social. Por ser un puesto que no tiene remuneración alguna, y/o contrato para ofrecer prestaciones de ley, encontramos el lamentable hecho de tener formas arcaicas y carentes de fiabilidad para el control de los mismos.

Los sistemas computacionales de hoy en día logran eliminar un trabajo adicional innecesario en muchos de los procedimientos donde se requiera realizar operaciones matemáticas iterativas, o bien simplemente llevar un registro perfecto de toda una multitud de usuarios.

En la Facultad de Estudios Superiores Aragón existe una programa de computadora que contabiliza el tiempo de labor de los prestadores de servicio social, el problema es que ha sido implementada de una manera informal, quedando así como beneficio de los universitarios que realizan su servicio exclusivamente en el departamento de informática y en las salas de Fundación UNAM.

Todas las demás áreas de la Universidad no tienen un sistema que controle dicha variable, por ejemplo: el SUA, el Centro de Investigación, el Centro de Lenguas Extranjeras (CLE), el Centro de Computo, las jefaturas de cada una de las carreras que se imparten en esta facultad, entre otras; éstas se encargan de realizar el proceso de administración de sus prestadores de servicio social anotando con bolígrafo en alguna libreta teniendo que realizar el conteo a mano y calcular cuándo finaliza el servicio.

Grosso modo, el “Sistema de Conteo para prestadores de Servicio Social en la FES Aragón (SICSS)” es en esencia una página web, está desarrollada de tal modo que luce como una aplicación común de escritorio, por ejemplo: el procesador de textos en donde se ha realizado este proyecto (Microsoft® Word) es un paquete o aplicación de escritorio, fácil de comprender y utilizar por cualquier ser humano.

Cabe aclarar que una página web es un sistema gráfico de hipertexto almacenado en un computador con características especiales llamado “servidor”, al cual se puede ingresar desde otras computadoras “clientes” gracias a internet.

Dicho programa tiene limitaciones, tales como no poder realizar conexiones a servidores remotos, esto significa que estos programas sólo funcionan dentro de una misma computadora sin la opción de poder leer los datos desde cualquier otro lugar sin necesidad de acudir al mismo sitio donde se encuentra el equipo de cómputo, además de no contar con historiales de registros o restricción de computadoras, entre otras herramientas, para aumentar la seguridad del sistema.

La inseguridad mayor de dichos sistemas radica en la posible falla de algún dispositivo de la computadora, además de tener la posibilidad de perder toda la información en algún momento inesperado sin alternativa de recuperación.



La metodología a utilizar para la realización del SICSS se basa en la técnica de casos de uso del Lenguaje Unificado de Modelado (UML), de este modo se podrá detallar cada una de las secciones que conforman la página, y posteriormente pueda ser visualizado a través de un diagrama que ilustre el flujo de trabajo empleado por cada uno de los partícipes del sistema.

El SICSS está diseñado con base en la última generación de computadoras, para su creación se han empleado los siguientes lenguajes de programación de código abierto (gratuitos) y mayor auge dentro del mundo Web:

PHP.- Tienen la función de realizar los procesos internos del sistema Web del lado del servidor.

JavaScript.- Se encargarán de hacer dinámica e interactiva la página (más fácil de usar) del lado del cliente o navegador.

PostgreSQL.- Se ha utilizado para almacenar y gestionar toda la información de cada uno de los usuarios.

CSS, Adobe® Photoshop® y Adobe® Flash®.- Se emplean para crear un diseño innovador, atractivo y amigable a la vista del usuario.

XML.- Para manejar listas de datos que comúnmente son repetitivas, en un mismo archivo.

AJAX.- Tendrá poca funcionalidad esta técnica de programar dentro del SICSS, pero su uso es principalmente para enviar y recibir datos en la página web sin necesidad de recargarla por completo.

El objetivo general es implementar el sistema de cómputo SICSS para llevar un registro perfecto de los prestadores de servicio social, con el fin de agilizar y mejorar ese procedimiento en la Facultad de Estudios Superiores Aragón.

Con ello se facilitaría la administración y control de los prestadores de servicio, se lograría consultar el tiempo exacto de servicio desde cualquier computadora, mediante internet. Así como evitar fraudes o modificaciones no deseadas en el conteo del tiempo de cada servicio social registrado.

Por otro lado se pretende explotar los recursos de la universidad aprovechando la gran capacidad y número de los equipos de cómputo para aumentar la competitividad de la Facultad de Estudios Superiores Aragón con otras instituciones.

El realizar un proyecto de este tipo y deducir si es lo mejor para una organización no es complicado de entender, ya que en muchas instituciones a nivel mundial se desarrollan, mejoran y aplican nuevos sistemas a diario, desde una pequeña empresa, como alguna miscelánea, que tiene un pequeño programa de computadora para automatizar el

procedimiento de cobro y verificación de precios; hasta una gran organización donde no sólo existe un sistema capaz de administrar productos, sino que se desarrollan día a día muchos otros sistemas que mejoren e innoven cada una de las tareas que se realizan, y todos ellos con la misma finalidad, automatizar procesos y minimizar el trabajo humano.

Teniendo en cuenta que México no es un país de primer mundo, es muy claro que éste tipo de ideas son las que día a día nos harán más competitivos y qué mejor si hacemos nuevos sistemas informáticos aplicados a instituciones grandes y prestigiadas, como lo es la UNAM. De esta manera optimizamos las funciones e innovamos los servicios de la institución.

Entonces, el objetivo específico de implementar un sistema que administre la permanencia temporal del alumno o egresado en su servicio social dentro de la FES Aragón, es que los usuarios tengan una interfaz que les sea más amigable y sencilla al realizar el registro de su tiempo en labor, así como familiarizarlo más con los equipos de cómputo, tomando en cuenta que no todos tienen esa facilidad; con la opción de consultar sus horas acumuladas en cualquier momento y desde cualquier lugar. Al mismo tiempo ayudará a los asesores a manejar un mayor número de universitarios sin preocuparse por algún tipo de error, llevando un mejor control de los mismos, así como también se conseguirá eliminar el tiempo que normalmente se invertía en dar de alta o baja a cada alumno por su propia mano.

Todos los programas orientados a bases de datos tienen la característica de trascender muy fácilmente, lo que significa que los datos pueden ser almacenados por un tiempo totalmente indefinido, además de ser muy sencillo migrar información o respaldarla para evitar pérdidas de ésta.

La información actualmente es muy sencilla de copiar y transportar, se puede hacer con un simple dispositivo de almacenamiento extraíble, tal como un disco compacto, una memoria USB u otra computadora a través de una red. Esto lleva sin lugar a duda a un impacto exitoso dentro de la nueva generación de los sistemas computacionales sin temor al fracaso. Este tipo de proyectos son totalmente trascendentales, ya que gracias a su gran versatilidad y facilidad de manejo, puede extenderse a otras Facultades o Universidades con ayuda de algún programador web.

Afortunadamente la FES Aragón cuenta con excelente equipo de cómputo y la cantidad necesaria de servidores para tener un servicio muy competitivo a nivel Universidad; por lo tanto es apropiado explotar dichos recursos por bien de la Facultad, iniciando con la creación y posteriormente implementación de sistemas que minimicen el trabajo del personal. Los alcances de este proyecto se facilitan al tener estas ventajas de recursos materiales y el no tener la necesidad de requerir mayor disponibilidad financiera, ya que el sistema no requiere adquirir ningún recurso adicional.

Actualmente la facilidad de manejo de una computadora, a nivel superior, es muy básica para aquellas licenciaturas que no están muy relacionadas con el ámbito tecnológico, sin embargo día a día va creciendo la necesidad de aprendizaje y la relación con las máquinas. Con esto debemos contemplar que algunos usuarios no puedan utilizar fácilmente el sistema, sin embargo esto no es problema, no solo porque la interfaz es muy sencilla, sino porque la disponibilidad de becarios en la Facultad será la herramienta más útil para ayudar a la comprensión del usuario, en cuanto al procedimiento, y ésta no creará desgaste de tiempo sino que únicamente será una inversión inicial que posteriormente no se volverá a realizar; no es una capacitación la que se ofrecerá, sólo una breve explicación tan llana como acceder a tu correo electrónico.

# CAPÍTULO 1

## FUNDAMENTOS DE COMPUTACIÓN

---

### 1.1 ANTECEDENTES

Las bases de funcionamiento de los sistemas de cómputo actuales se basan en diversas herramientas que se han llevado a un nivel ya muy avanzado. Dichas herramientas están generadas por todo un mundo de cálculos matemáticos, aunque estos a su vez están montados sobre sistemas binarios; esto significa que mediante operaciones aritméticas binarias, utilizadas previamente para dar funcionamiento a una computadora, se resuelven problemas matemáticos muy avanzados.

*Una computadora se constituye de varios componentes físicos o hardware que incluyen la unidad central de proceso, el teclado, el ratón, el monitor, la impresora y las unidades de disco. También existe la parte intangible que corresponde al lenguaje, los programas y datos, que se denomina software.<sup>1</sup>*

Estos son solo los componentes principales de un equipo de cómputo, de ahí se desprenden muchos otros dispositivos útiles, sin embargo para poder explicar los conceptos que llevaron a la creación de sistemas electrónicos digitales no hace falta citar cada uno de ellos.

---

<sup>1</sup> Fournier Julio, "Introducción a la informática", p. 27.

Uno de los componentes más importantes es el CPU, *ésta unidad es la principal y representa el cerebro de la computadora, aquí se contiene una unidad aritmética lógica que es donde se procesan los datos, principalmente operaciones lógicas como sumas y productos binarios*<sup>2</sup>.

Para conseguir introducir éste tipo de sistema llevó todo un estudio que remonta a los siglos XVII y XVIII donde los esfuerzos por crear mecanismos que fueran capaces de realizar cuentas más rápidas que el cerebro humano, se lograron. No obstante los alcances no cubrieron las expectativas del hombre, y después de crear máquinas mecánicas con sistemas analógicos que se constituyen de tarjetas perforadas y engranes, se lograron hacer equipos más completos que ya contenían transistores y diodos que daban una mejor funcionalidad y eficiencia a las máquinas. Pero para lograr llegar a esto se tuvo que desarrollar todo un sistema matemático que involucra desde operaciones sumatorias hasta las logarítmicas, afortunadamente este tipo de técnicas ya no son empleadas en la actualidad, ahora sólo nos basamos en operaciones binarias, ya que las computadoras digitales funcionan únicamente mediante este mecanismo.

En el siguiente apartado se tratará a detalle cada uno de los puntos mencionados anteriormente, así como la evidente evolución a través de las generaciones de computadoras, desde la máquina analítica de Charles Babbage hasta la sexta generación de computación que es en la que vivimos en la actualidad.

## 1.2 SEÑAL ANALÓGICA Y DIGITAL

Inicialmente se procedía a realizar procesos de manera analógica, con ayuda de papel y tinta; la tecnología ha transformado este sistema y lo ha nombrado digital, supliendo a lo analógico, *con la gran ventaja que los datos digitales se pueden procesar y transmitir de forma más eficiente y fiable que los datos analógicos*.<sup>3</sup>

Nuestro mundo desde siempre ha funcionado de manera analógica, es decir, todo el entorno contiene parámetros físicos, y a diferencia de una computadora, ésta funciona de forma digital; aunque también existen dispositivos de computación analógica los cuales tienen un mecanismo diseñado específicamente para explotar una analogía en tal forma que puedan obtenerse soluciones visuales o cuantitativas a los problemas en el

---

<sup>2</sup> Ibidem p. 28

<sup>3</sup> Thomas L. Floyd, "Fundamentos de sistemas digitales", p. 2,4.

campo de interés<sup>4</sup>, sin embargo, esta tecnología solo forma parte de un sistema cuyos resultados serán digitales.

*Los circuitos electrónicos se pueden dividir en: analógicos y digitales, donde las magnitudes analógicas son aquellas que toman valores continuos, mientras que las digitales tienen valores discretos<sup>5</sup>.*

Por ejemplo un ventilador dentro de un gabinete de computadora tiene un funcionamiento analógico, ya que los valores de su voltaje varían sucesivamente, es decir de manera continua, hasta alcanzar el valor deseado. Por otro lado el término digital se deriva de la forma en que las computadoras realizan las operaciones, a éstas señales se les llama discontinuas o discretas, es decir que su valor sólo podrá variar entre uno y cero (sistema binario); esta tecnología ha progresado desde los circuitos de válvulas de vacío, hasta los transistores y circuitos integrados.<sup>6</sup>

### 1.3 SISTEMA BINARIO

En el inicio de los sistemas analógicos ya existía el término binario, sin darle la importancia que con ello se pudieran crear mecanismos más potentes que los que tenían en esos tiempos, por ejemplo los simples interruptores sólo tienen dos posibles valores, uno o cero, donde “uno” representa el paso de corriente o el interruptor encendido, y “cero” la ausencia de corriente o apagado. La sencillez de este modelo matemático no llevó a generar ideas nuevas, sino que posteriormente ayudó a completarlas cuando ya se tenían sistemas más avanzados. A pesar de este avance, que ya podía ser considerado digital, aún se tardaron muchos años más para poder implementar este tipo de numeración dentro de circuitos de los que actualmente gozamos.

*Gottfried Wilhelm von Leibniz, fue un gran apasionado del sistema numérico binario, era un matemático alemán que creía que había algo de belleza en la analogía existente entre el 0, que representaba el vacío, y el 1, que representaba la divinidad<sup>7</sup>. Aunque algunos otros filósofos creían que las ideas de este matemático estaban orientadas a lo místico, sin embargo la complejidad del sistema se fue haciendo aun mayor aunque no lo parezca.*

<sup>4</sup> Leon Presser, “Ciencias de la computación”, p. 15.

<sup>5</sup> Thomas L. Floyd, Op cit p. 4

<sup>6</sup> Ibidem p. 5

<sup>7</sup> Bartee Thomas C., “Fundamentos de computadoras digitales”, p.53.

## 1.4 GENERACIONES DE COMPUTADORAS

El avance a la tecnología digital llegó algunas generaciones después de las invenciones analógicas, en los inicios solo eran máquinas mecánicas el ejemplo más claro viene desde la primera calculadora, donde la única operación que realizaba era la suma, *fue descubierta en el siglo XVII por el filósofo y matemático Blaise Pascal. Actualmente se ha perfeccionado y sofisticado el mecanismo, tanto en máquinas registradoras como en calculadoras de mesa, sin embargo contiene los mismos principios del inventor. A estas máquinas no se les puede llamar automáticas porque requieren de la continua intervención del operador para introducir nuevos datos. Este principio generó una gran idea al ingeniero Charles Babbage al crear una máquina analítica, la cual ya fue denominada calculador universal era capaz de almacenar distintos programas según un esquema en todo análogo a las computadoras electrónicas actuales*<sup>8</sup>.

Las primeras computadoras no tenían la misma eficiencia y capacidad que las actuales, sin embargo existen diferentes generaciones de computadoras para las cuales **ha existido un software que se adecúa a los alcances de su hardware**. A pesar que éste último se ha desarrollado de manera más lenta que los programas y lenguajes, se ha conseguido hasta el día de hoy un equilibrio considerable para soportar la cantidad diversa de software existente.

### 1.4.1 PRIMERA GENERACIÓN (1951 A 1958)

Las computadoras de la primera Generación emplearon bulbos para procesar información. Los operadores ingresaban los datos y programas en código especial por medio de tarjetas perforadas. El almacenamiento interno se lograba con un tambor que giraba rápidamente, sobre el cual un dispositivo de lectura/escritura colocaba marcas magnéticas. Esas computadoras de bulbos eran mucho más grandes y generaban más calor que los modelos contemporáneos.

Como resultado, el costo, volumen, consumo de fuerza, el retardo de lógica y la cantidad de fallas de cada red eran muy elevados comparados con las redes equivalentes semiconductoras de hoy<sup>9</sup>.

La IBM tenía monopolizado por completo toda la red de computadoras que utilizaban tarjetas perforadas, por lo que su gran impacto en el mercado fue lo que le dio auge en esa época, además tenían productos como relojes, rebanadores de carne, básculas, entre otros, que los llevaron a darse a conocer aún más. Entonces comenzaron a construir computadoras electrónicas y la empresa IBM instaló arriba de 1000

<sup>8</sup> Aréchiga Rafael, "Fundamentos de computación", pp.3-9.

<sup>9</sup> Leon Presser, Opcit p.371.

computadoras en E.U, aunque caras y de uso limitado las computadoras fueron aceptadas rápidamente por las Compañías privadas y de Gobierno. A la mitad de los años 50 IBM y Remington Rand se consolidaban como líderes en la fabricación de computadoras.

Los lenguajes de programación que surgieron con este desarrollo tecnológico fueron:

- o Plankalkül
- o ENIAC
- o C-10
- o Autocode
- o FORTRAN

## 1.4.2 SEGUNDA GENERACIÓN (1959-1964)

### 1.4.2.1 Transistores

El invento del transistor hizo posible una nueva Generación de computadoras, más rápidas, más pequeñas y con menores necesidades de ventilación, ya que vinieron a sustituir a los bulbos. *Este dispositivo fue inventado por los norteamericanos J. Barden, M. Brattain y W. Shockely. Básicamente el funcionamiento radica de acuerdo a ciertos valores de tensión eléctrica que se le apliquen al dispositivo, éste conduce o no corriente eléctrica y representa así el "1" o el "0", necesarios para el funcionamiento de las computadoras<sup>10</sup>.*

Sin embargo el costo seguía siendo una porción significativa del presupuesto de una Compañía. Las computadoras de la segunda generación también utilizaban redes de núcleos magnéticos en lugar de tambores giratorios para el almacenamiento primario. Estos núcleos contenían pequeños anillos de material magnético, enlazados entre sí, en los cuales podían almacenarse datos e instrucciones.

Los programas de computadoras también mejoraron. El COBOL (COMmon Business Oriented Lenguaje) desarrollado durante la 1era generación estaba ya disponible comercialmente, éste representa uno de los más grandes avances en cuanto a portabilidad de programas entre diferentes computadoras; es decir, es uno de los primeros programas que se pueden ejecutar en diversos equipos de computo después de un sencillo procesamiento de compilación. Los programas escritos para una computadora podían transferirse a otra con un mínimo esfuerzo.

Grace Murria Hooper (1906-1992), quien en 1952 había inventado el primer compilador fue una de las principales figuras de CODASYL (Comité on Data SYstems Languages), que se encargó de desarrollar el proyecto COBOL. El escribir un programa ya no

---

<sup>10</sup> Lorenzo Francisco, "Introducción a la computación a través de logo", pp. 14, 15.



requería entender plenamente el hardware de la computación. Las computadoras de la 2da Generación eran sustancialmente más pequeñas y rápidas que las de bulbos, y se usaban para nuevas aplicaciones, como en los sistemas para reservación en líneas aéreas, control de tráfico aéreo y simulaciones para uso general. Las empresas comenzaron a aplicar las computadoras a tareas de almacenamiento de registros, como manejo de inventarios, nómina y contabilidad.

Los lenguajes de programación más sobresalientes que nacieron en esta generación son los siguientes:

- o LISP
- o ALGOL
- o COBOL
- o APL
- o Simula

### 1.4.3 TERCERA GENERACIÓN (1964-1971)

#### Circuitos Integrados

*Las computadoras de la tercera generación emergieron con el desarrollo de los circuitos integrados (pastillas de Silicio) en las cuales se colocan miles de componentes electrónicos, en una integración en miniatura. Las computadoras nuevamente se hicieron más pequeñas, más rápidas, desprendían menos calor y eran energéticamente más eficientes, aunque la diferencia entre la segunda y esta generación no es tan notoria como lo fue la evolución del bulbo al transistor<sup>11</sup>.*

Antes de la existencia de los circuitos integrados, las computadoras estaban diseñadas para aplicaciones matemáticas o de negocios, pero no para las dos cosas. Los circuitos integrados permitieron a los fabricantes de computadoras incrementar la flexibilidad de los programas, y estandarizar sus modelos.

Se empiezan a utilizar los medios magnéticos de almacenamiento, como cintas magnéticas y enormes discos rígidos. Algunos sistemas todavía usan las tarjetas perforadas para la entrada de datos, pero las lectoras de tarjetas ya alcanzan velocidades respetables.

Las computadoras trabajaban a tal velocidad que proporcionaban la capacidad de correr más de un programa de manera simultánea (multiprogramación).

Aquí surgieron lenguajes de programación como:

- o Basic

---

<sup>11</sup> Lorenzo Francisco, Opcit pp. 16,17.

- o PL/I
- o BCPL
- o B
- o Pascal

#### 1.4.4 CUARTA GENERACIÓN (1971 A 1981)

##### Microprocesador y Chips de memoria

Dos mejoras en la tecnología de las computadoras marcan el inicio de la cuarta generación: el reemplazo de las memorias con núcleos magnéticos, por las de chips de Silicio y la colocación de muchos más componentes en un Chip: producto de la micro miniaturización de los circuitos electrónicos. El tamaño reducido del microprocesador y de chips hizo posible la creación de las computadoras personales (PC).

Actualmente ha surgido una enorme cantidad de fabricantes de microcomputadoras o computadoras personales, que utilizando diferentes estructuras se pelean por el mercado de la computación, el cual ha llegado a crecer tanto que es uno de los más grandes a nivel mundial; sobre todo, a partir de 1990, cuando se logran sorprendentes avances en Internet.

Esta generación de computadoras se caracterizó por grandes avances tecnológicos realizados en un tiempo muy corto.

Los sistemas operativos han alcanzado un notable desarrollo, sobre todo por la posibilidad de generar gráficos a grandes velocidades, lo cual permite utilizar las interfaces gráficas de usuario (Graphic User Interface, GUI), que son pantallas con ventanas, iconos (figuras) y menús desplegables que facilitan las tareas de comunicación entre el usuario y la computadora, tales como la selección de comandos del sistema operativo para realizar operaciones de copiado o formato con una simple pulsación de cualquier botón del ratón (mouse) sobre uno de los iconos o menús.

Otro aspecto importante en esta generación es la implementación de un sistema operativo, MS- DOS (MicroSoft Disk Operating System), con el cual surgieron ideas de programación más avanzada. *Los nuevos lenguajes fueron orientados a la "inteligencia artificial", como:*

- o C
- o Prolog
- o ML
- o LOGO
- o SmallTalk
- o SQL

En esta generación apareció por fin el tan actualmente utilizado SQL (Structured Query Language), el cual está orientado a diseño de bases de datos, más adelante se tratará este apartado a fondo.

#### **1.4.5 QUINTA GENERACIÓN (1982-1989)**

##### **Inteligencia artificial**

La identificación de las generaciones de computadoras cada vez se hace más compleja, porque los grandes avances y nuevos descubrimientos ya no son tan sorprendentes como lo fueron en un inicio. Hay algunos autores que consideran que la cuarta y quinta generación han terminado, y las ubican entre los años 1971 y 1990. Ellos consideran que la sexta generación está en desarrollo desde 1990 hasta la fecha.

Siguiendo la pista a los acontecimientos tecnológicos en materia de computación e informática, podemos puntualizar algunas fechas y características de lo que podría ser la quinta generación de computadoras.

Los grandes acontecimientos tecnológicos en materia de microelectrónica y computación, inteligencia artificial, sistemas expertos, redes neuronales, teoría del caos, algoritmos genéticos, fibras ópticas, telecomunicaciones, etc. establecieron las bases de lo que se puede conocer como quinta generación de computadoras.

Aunque en teoría el trabajo con varios microprocesadores debería ser mucho más rápido, es necesario llevar a cabo una programación especial que permita asignar diferentes tareas de un mismo proceso a los diversos microprocesadores que intervienen.

También se debe adecuar la memoria para que pueda atender los requerimientos de los procesadores al mismo tiempo. Para solucionar este problema se tuvieron que diseñar módulos de memoria compartida capaces de asignar áreas de caché para cada procesador.

Durante esta generación se desarrollaron sistemas expertos, los cuales se dice que razonan, así como la creación de programas dedicados al reconocimiento del lenguaje natural, o el reconocimiento de imágenes; *se dice que un sistema experto tiene la capacidad de razonar porque posee un alto nivel de experiencia humana en un terreno muy particular, además se ha conseguido transferir aspectos específicos de la inteligencia humana a los ordenadores<sup>12</sup>.*

---

<sup>12</sup> Bishop Peter, "Computadores de la quinta generación", p.36.

El almacenamiento de información se realiza en dispositivos magneto ópticos con capacidades de decenas de Gigabytes; se establece el DVD (Digital Video Disk o Digital Versatile Disk) como estándar para el almacenamiento de video y sonido; la capacidad de almacenamiento de datos comenzó a crecer de manera exponencial. Los componentes de los microprocesadores actuales utilizan tecnologías de alta y ultra integración, denominadas VLSI (Very Large Scale Integration) y ULSI (Ultra Large Scale Integration).

Los lenguajes de programación con mayor auge dentro de esta generación son:

- o Ada
- o C++
- o Eiffel
- o Pearl
- o FL
- o Haskell

Además con esta generación viene la aparición del Windows 95 y posteriormente Windows 98 que revolucionaron la tecnología de programación orientada a gráficos, donde el internet comienza a extenderse gracias a la introducción de los correos electrónicos.

#### **1.4.6 SEXTA GENERACIÓN (1990 HASTA LA FECHA)**

Inicialmente cabe aclarar que existen diversas bibliografías que varían entre los años de las últimas generaciones de computadoras, sin embargo se puede hacer un esbozo de los principales factores que representan a estas últimas generaciones.

Las computadoras de la sexta generación cuentan con arquitecturas combinadas Paralelo / Vectorial, con cientos de microprocesadores vectoriales trabajando al mismo tiempo; se han creado computadoras capaces de realizar más de un millón de millones de operaciones aritméticas de punto flotante por segundo (teraflops); las redes de área mundial (Wide Area Network, WAN) seguirán creciendo desorbitadamente utilizando medios de comunicación a través de fibras ópticas y satélites, con anchos de banda impresionantes. Las tecnologías de esta generación ya han sido desarrolladas o están en ese proceso. Algunas de ellas son: inteligencia / artificial distribuida; teoría del caos, sistemas difusos, holografía, transistores ópticos, entre otros.

Pero existe una limitante entre diversas tecnologías que dan pie a separar una generación de otra; la más evidente sin duda es la tecnología para visualizar los resultados de una computadora, es decir, las pantallas LCD y de plasma, las cuales consumen menos energía que los antiguos monitores y dan una resolución de video con mucha mayor calidad. Otro aspecto importante es que el software para computadoras y

los sistemas operativos comienzan a formar parte de todo un mercado de servidores, orientado a tecnologías en red y claro, de internet; la edición de audio y video, visualización de televisión y hasta emulación de consolas de videojuegos son solo aspectos que emergieron desde el surgimiento de tan alta tecnología, diciéndose que ya pertenecen a una “séptima generación”, la cual abarcaría de finales de los 90 hasta nuestros días.

Los lenguajes de programación que actualmente son más utilizados y nacieron junto con estas últimas generaciones son los siguientes:

- o Python
- o Ruby
- o UAP
- o C#
- o Java
- o PHP
- o CSS
- o JavaScript
- o MySQL
- o PostgreSQL
- o AJAX

## 1.5 COMPILACIÓN E INTERPRETACIÓN

Un aspecto de interés para los siguientes apartados es que no todos los lenguajes son iguales, es decir, que cada uno tiene una estructura diferente y están orientados a diferentes ámbitos, *éstos pueden ser lenguajes de ensamble o de procedimiento, los de ensamble (ensambladores) se utilizan abreviaturas mnemotécnicas y las posiciones de memoria, lo cual es algo complejo para programar, y a diferencia de los de procedimiento, son más universales y fáciles de usar, ya que no requieren descomposición de fórmulas y las instrucciones vienen directamente en inglés en lugar de siglas mnemotécnicas.*<sup>13</sup>.

Para conseguir escribir un programa en lenguajes de procedimiento es necesario apoyarnos en un *compilador, este programa descompone cada instrucción en pasos elementales y asigna direcciones numéricas, creando un archivo llamado “objeto” el cual está escrito en código máquina, es decir, solo unos y ceros*<sup>14</sup>. Un compilador tiene la

<sup>13</sup> Stoutemyer David R., “Programación para ciencia e Ingeniería”, p. 8.

<sup>14</sup> Ibidem p. 9.

capacidad de detectar si existe error alguno dentro de todo el *código fuente*<sup>15</sup> y si lo hay, el programa procede a pedirle al programador que corrija el error antes de ser ejecutado por la máquina, además que no requerimos escribir posiciones de memoria ya que él mismo lo hace.

Por otro lado un intérprete (ejecución), analiza el programa fuente y lo ejecuta directamente sin importar si existen errores, obviamente el tiempo de ejecución es menor que con los compiladores, aunque la garantía de que los resultados sean correctos es poca. Sin embargo una ventaja viable es que se pueden reparar los errores sin necesidad de recargar el código.

Además de los tipos de lenguajes, éstos pueden estar orientados al diseño de programas sobre un sistema operativo (programas de usuario), a programar el sistema operativo mismo (programas de sistema y virus), orientados a internet (diseño Web) y al almacenamiento masivo de datos (Bases de Datos).

## 1.6 NAVEGADORES DE INTERNET

*Un navegador o navegador web (del inglés, web browser) es un programa que permite ver la información que contiene una página web, (ya se encuentre ésta alojada en un servidor o dentro de la World Wide Web*<sup>16</sup>.

Un navegador no necesariamente depende de internet para verlo funcionar, éste puede mostrar el contenido del código HTML contenido en un archivo, lo que lo convierte en un programa que interpreta código y lo visualiza como una página. Sin embargo su creación fue dedicada a la tecnología desarrollada para interactuar con diversas redes a lo largo del mundo, para así crear una red de redes, o internet.

En un navegador común como “Internet Explorer ®”, para ingresar a una página local se requiere la dirección 127.0.0.1 por estándar, pero para entrar a otro servidor del mundo se requiere su dirección propia ó IP, por ejemplo para entrar a la ya famosa empresa “Google” se puede digitar la dirección <http://74.125.19.104/> pero gracias al protocolo del Dominio de Nombres del Sistema (DNS) es posible entrar a la página con la dirección <http://www.google.com>.

Existen diversos lenguajes orientados a la programación para creación de páginas Web, sin embargo el padre e iniciador de este tipo de programación es sin duda el HTML.

---

<sup>15</sup> Programa en lenguaje de procedimiento, previo al código objeto.

<sup>16</sup> [http://es.wikipedia.org/wiki/Navegador\\_web](http://es.wikipedia.org/wiki/Navegador_web).

Para establecer una conexión a internet cabe mencionar que existen protocolos encargados de direccionar y controlar a todos los servidores del mundo que manejan alguna página Web, éstos protocolos son los ya famosos HTTP (Protocolo de transferencia de Hipertexto) y WWW (Red Global Mundial), dónde *servidor* es una computadora con características especiales y contenedor de los archivos ya programados con los diversos lenguajes web existentes, entre ellos HTML, para que el navegador pueda interpretarlos y posteriormente sean visualizados por el usuario.

La forma de manipular un navegador usando estos protocolos se hace mediante una URL (Uniform Resource Locator), ésta se encarga de llevar al usuario a través de una dirección única (IP) que identifica al servidor, en dónde se traducirán los archivos que conforman la página web, como ya vimos en el ejemplo de Google.

Una IP (Internet Protocol) es simplemente una cadena de números clasificados de tal modo que sea imposible que alguno se repita en el mundo, por ejemplo el servidor que alojará a mi proyecto SICSS tiene asignada la IP: "132.248.221.198".

Existe una gran diversidad de navegadores de internet, sin embargo los más utilizados en la actualidad debido a su buen nivel de seguridad e interpretación, según mi experiencia son: Internet Explorer, Mozilla Firefox, Google Chrome, Safari y Opera.

## CAPÍTULO 2

# DESARROLLO WEB

---

---

Para desarrollar cualquier aplicación o sistema web se requiere pensar en la forma en que el ordenador o computadora entienda dicho proyecto, es decir, un lenguaje de programación. Sin embargo, la utilización de ellos no se puede introducir de inmediato en la fase de diseño debido a su rigidez.

Es por eso que para este proyecto se ha empleado el uso de diversas herramientas para la planificación y análisis de un “**S**istema de **C**onteo para **S**ervicios **S**ociales (SICSS)”, tales como los casos de uso del Lenguaje Unificado de Modelado (UML), diagramas de casos de uso, diagramas de estado y diagramas de secuencia.

*De este modo se pueden hacer representaciones gráficas consiguiendo un lenguaje intermedio entre el lenguaje natural y el de programación, de tal manera que permita flexibilidad para expresar las acciones que se han de realizar, para así no depender de un lenguaje de programación durante el diseño y posteriormente obtener el algoritmo para codificarlo en el lenguaje que nos interese<sup>17</sup>.*

---

<sup>17</sup> Alcalde Eduardo, “Metodología de la programación”, p.49



## 2.1 LENGUAJES DE PROGRAMACIÓN WEB

### 2.1.1 HTML

Para realizar un programa orientado a internet, mejor conocido como página Web, en sus inicios se empleaba únicamente el lenguaje HTML el cual representa el código base dentro de la red de redes, mejor conocida como internet.

No obstante los programadores analistas dedicados al desarrollo de software, implementaron también rutinas de código que hicieran más interactiva una página web, estos avances llevaron a la creación de herramientas complementarias de programación Web, tales como: PHP, Java Script, CSS y XML entre otros.

HTML es un lenguaje de marcado. Esto significa que no es un lenguaje de alto nivel como Java o C. *En lugar de compilarse y ejecutarse, HTML es leído, o interpretado, por un agente usuario. A este agente se le conoce como navegador Web*<sup>18</sup>.

Grosso modo para crear una página en HTML, primero se debe realizar un bosquejo de lo que se quiere hacer, con base en todo un estudio y análisis de los requerimientos que necesite el cliente para hallar la solución más viable mediante un diseño. Éste diseño puede crearse mediante el sistema UML (Unified Modeling Language ó Lenguaje Unificado de Modelado), o sencillamente con diagramas de flujo o a bloques; esto depende de la técnica que desee emplear el programador para crear el sistema.

Una vez que se sabe lo que se quiere programar se procede a crear los archivos con el código fuente, los cuales serán interpretados mediante un navegador Web (internet explorer, netscape, mozilla firefox, opera, safari y google chrome, de entre los más comunes).

### 2.1.2 JAVASCRIPT

*El lenguaje Java Script es un lenguaje de programación avanzada, basado en objetos, para el diseño de páginas Web interactivas*<sup>19</sup>, éste sirve como complemento para otros lenguajes como HTML y PHP, sin embargo a pesar de ser un intérprete de alto nivel, no se compara con lenguajes orientados a objetos de manera avanzada como los son C++ ó Java.

JavaScript tiene una pequeña desventaja, que actualmente para muchos ya no lo es, se trata de la dependencia hacia las características del navegador Web que se esté

<sup>18</sup> Holzschlag Molly E., "La biblia de HTML 4", p.36.

<sup>19</sup> Bobadilla S. Jesús, "HTML dinámico, ASP y JavaScript a través de ejemplos", p. 1

utilizando, esto es que no todos los programas para navegar por internet son compatibles con los scripts (códigos) de JavaScript. Sin embargo en la actualidad ya se están estandarizando todos los exploradores, reglas muy similares de interpretación de código.

Existen otros lenguajes que extienden las capacidades de una página HTML para hacerla más dinámica, tal es el caso de CGI, pero este lenguaje es más difícil de aprender y utilizar. JavaScript es más flexible que cualquier otro lenguaje dedicado al dinamismo Web, además tiene la ventaja de poder mostrar resultados instantáneos en una Web sin tener que esperar a que responda el servidor remoto, ya que JavaScript se encarga de introducir código extra para respuestas veloces, por ejemplo en formularios en donde exista un número limitado de opciones, se puede hacer un pequeño menú con dichas alternativas, y con el uso simple de HTML tendría que enviarse un parámetro al servidor que tuviese que responder con el menú de respuestas apropiado.

En resumen, JavaScript ayudará a darle un entorno interactivo a la página, aumentar la flexibilidad para su uso y hacerla más atractiva para los usuarios.

### 2.1.3 PHP

*PHP (Hypertext Pre-Processor) es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Está orientado principalmente al desarrollo de aplicaciones Web para diseñar páginas dinámicas de servidor capaces de responder de manera inteligente a las peticiones del cliente, con la capacidad de automatizar un sinnúmero de tareas<sup>20</sup>.*

Éste lenguaje es relativamente nuevo, nacido en 1994 y actualmente usado junto con otros lenguajes como ASP, JSP, ColdFusion y Perl, orientados al trabajo directo sobre servidores.

PHP es un programa de utilización masiva ya que tiene características notablemente favorables, tales como ser un software libre (utilización gratuita), multiplataforma (Win32, Unix, Mac OS X, Linux, etc.) y sobre todo tener un nivel de dificultad de programación muy bajo, manteniendo los resultados que tendría uno avanzado, como conexiones con bases de datos y generación de gráficos, entre otros.

También tiene la capacidad de soportar diversos servidores Web, como Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Caudium, Xitami, entre muchos otros; y más de 25 gestores de bases de datos, entre los más comunes están: Oracle, dBase, Sybase, MySQL y PostgreSQL. Así como la compatibilidad con todo un mundo de servicios, protocolos, objetos, extensiones y toda una gama de procedimientos que van en crecimiento cada día.

---

<sup>20</sup> Gutiérrez R. Abraham, "PHP 5 a través de ejemplos", p. XV

Las versiones en este ámbito son fundamentales ya que cada una trabaja con procedimientos diferentes, corrección de errores, implementación de nuevas bases de datos o modificación de las mismas, nuevas extensiones y compatibilidad con navegadores. La versión de PHP a utilizar para el SICSS es la más actual estable, es decir, PHP 5.

Otra gran ventaja de utilizar este lenguaje para gestionar la página en el servidor es que *PHP se usa, principalmente, para desarrollar contenido HTML dinámico, aunque puede usarse también para generar documentos XML (entre otros)*<sup>21</sup>.

En conclusión, el lenguaje PHP servirá como motor principal para la creación del sistema ya que todos los procesos internos del sistema estarán realizados sobre él. Además la conexión con el gestor de bases de datos, los esquemas HTML y CSS junto con los procesos ejecutados en el cliente como JavaScript y AJAX son controlados ó mandados a llamar desde PHP.

#### 2.1.4 CSS

La forma ideal de conseguir una página de internet con una gran variedad de estilos y la carga rápida de la misma es sin duda la utilización de la tecnología CSS.

*Las hojas de estilo en cascada o CSS (Cascading Style Sheets) se esfuerzan por alcanzar la separación completa del contenido y la presentación. En otras palabras, puedes construir, mantener y almacenar el contenido básico de tu página web, sitio o aplicación (texto, imágenes, formas, etc.) separándolo de la presentación visual (tales como el diseño, la tipografía y las imágenes decorativas)*<sup>22</sup>.

Al introducir un lenguaje como este también se debe considerar el uso apropiado del HTML ya que se depende de él por completo. La verdad es que es muy sencillo trabajar con estas dos herramientas, una vez entendida la forma en que el CSS manda a llamar a una parte del código HTML a través de un identificador único (id) es posible hacer cualquier cosa con las sencillas funciones que maneja este lenguaje de estilos.

Por ejemplo si se piensa en implementar un menú que tenga cierto desvanecimiento en sus bordes, una alineación especial, una transparencia o alpha, un tipo de letra, tamaño y la muestra de alguna pequeña imagen cuando se desliza el mouse, sólo se requiere una línea con no mas de 5 palabras de código por cada variable a utilizar, y así se puede mandar a llamar una y otra vez esa rutina CSS en cualquier instancia de la página web, con sólo atribuir el "id" en el lugar indicado del HTML.

---

<sup>21</sup> López Quijado José, "Domine PHP 5", p.1

<sup>22</sup> Croft Jeff, "Pro CSS Techniques", p.1

Para el SICCS se ha pensado utilizar en gran parte esta forma de programar ya que las actualizaciones de tal lenguaje, junto con las de los navegadores, van de la mano para así lograr mostrar el contenido de cualquier página que se haya desarrollado actualmente, y así conseguir un sistema que en diseño valla a la vanguardia.

### 2.1.5 XML

*El lenguaje extensible de etiquetas (eXtensible Markup Language) o XML no es un lenguaje de marcado como HTML sino un meta-lenguaje que nos permite definir lenguajes de marcado adecuado a usos determinados<sup>23</sup>. Se le llama metalenguaje porque sirve para crear lenguajes, y no es uno como tal.*

A las páginas que tiene montados documentos XML sobre el HTML se les conoce como XHTML, y para que éstas sean válidas requieren pasar por una especie de autenticación llamada DTD (Document Type Definition) o definición de tipo de documento, dicha plantilla establece la estructura adecuada de un documento XML.

La implementación de éste meta-lenguaje será útil para generar una lista de datos extraídos de PostgreSQL y que posteriormente sean leídos por PHP y así evitar leer los mismos datos una y otra vez, de este modo se consigue minimizar el uso del CPU. Aunque por otro lado se debe destacar su gran importancia al momento de implementar el tipo de programación AJAX que se verá en el siguiente apartado.

## 2.2 BASE DE DATOS POSTGRESQL

Una vez estructurado un programa que dé vida a una página en internet, debemos pensar en la administración de los datos que estaremos manejando. Cualquier sistema que se dedique al control de mucha información mediante un programa de computación, ya sea éste orientado a computadoras de escritorio o a servidores, debe tener un manejador de base de datos para su óptimo funcionamiento.

Tal y como se mencionó en las generaciones de computadoras, también la forma en que se administran los datos ha ido evolucionando, ya que la información de todo el mundo crece día a día de una manera impresionante, por lo tanto se requiere de gestores de bases de datos más veloces y eficaces, tanto para realizar registros, borrarlos y buscarlos, entre algunas otras opciones.

---

<sup>23</sup> Ibidem p.235

El gestor PostgreSQL está basado en el lenguaje madre de todas las bases de datos, SQL (**S**tructured **Q**uery **L**anguage – Lenguaje Estructurado de Consulta), orientado a bases de datos relacionales y consultas de datos.

*La estructura fundamental del modelo relacional es precisamente esa, "relación", es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos)<sup>24</sup>.*

Características de las bases de datos relacionales:

- . Una base de datos relacional se compone de varias tablas o relaciones.
- . No pueden existir dos tablas con el mismo nombre.
- . Cada tabla es a su vez un conjunto de registros, filas o tuplas.
- . Cada registro representa un objeto del mundo real.
- . Cada una de estos registros consta de varias columnas, campos o atributos.
- . No pueden existir dos columnas con el mismo nombre en la una misma tabla.
- . Los valores almacenados en una columna deben ser del mismo tipo de dato.
- . Todas las filas de una misma tabla poseen el mismo número de columnas.
- . No se considera el orden en que se almacenan los registros en las tablas.
- . No se considera el orden en que se almacenan las tablas en la base de datos.
- . La información puede ser recuperada o almacenada por medio de sentencias llamadas "consultas".

*PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado<sup>25</sup>.*

Las licencias BSD se encargan de establecer derechos de autor así como permitir la utilización del software sin limitación alguna. Por otro lado la utilización de Multiprocesos dentro de PostgreSQL evita que el sistema deje de funcionar cuando existen fallas en algún proceso, lo cual lo hace muy estable al estar operando.

Los lenguajes principales de compatibilidad para PostgreSQL son:

- \* PL/PgSQL
- \* C/C++.
- \* Gambas.
- \* Java PL/Java web.
- \* PL/Perl.

<sup>24</sup> [http://www.programacion.com/articulo/modelo\\_relacional\\_172](http://www.programacion.com/articulo/modelo_relacional_172)

<sup>25</sup> [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)

- \* PHP.
- \* PL/Python.
- \* PL/Ruby.
- \* PL/sh.
- \* PL/Tcl.
- \* PL/Scheme.
- \* PL/R.

## 2.3 AJAX

Para este apartado se debe destacar que AJAX, más que un lenguaje, es una técnica para programar a través de Javascript y apoyándonos en una codificación JSON (JavaScript Object Notation) interactuando con archivos XML, de sus siglas “Asynchronous Javascript and XML” logrando hacer peticiones al servidor como consultas a la base de datos y consiguiendo mostrar los resultados sin necesidad de recargar toda la página.

En la época actual AJAX es la forma más utilizada para desarrollar contenido web al gran potencial que consigue que una página se asimile a una aplicación común de escritorio.

Inicialmente cuando solo se trabajaba con los lenguajes de tipo cliente como HTML, Javascript y CSS, se le conocía al internet como la “web 1.0”, ahora nos encontramos con un avance dentro de este gran mundo al que se ha llamado la “Web 2.0”, *esto no es internet 2 sino una red paralela nueva de alta velocidad que funciona sobre la misma red de internet que todos conocemos*<sup>26</sup>.

El objetivo principal de este proyecto no es el de desarrollar un estudio sobre las nuevas tecnologías web ya que eso llevaría más de un solo proyecto, sino contemplar cada uno de los lenguajes, herramientas y técnicas empleadas para la construcción de una página web interactiva de vanguardia, explicando del modo mas simple como es que se logra tal efecto ante un usuario que desconoce por completo del tema.

Trabajar de manera asíncrona con XML y Javascript resulta bastante complejo, no solo por la individualidad e inclusión de librerías y frameworks adicionales (*la mayoría tiene la funcionalidad similar e incorpora algunas funciones propias con distintos grados de uso y*

---

<sup>26</sup> Firtman Maximiliano R., “AJAX Web 2.0 para profesionales”, p.3

*metodología de implementación*<sup>27</sup>), sino por la forma de codificar tu web ya que AJAX te traslada totalmente a otro modo de programar.

Para el caso del SICSS se empleara muy poco esta técnica por diversas causas, una es que llevaría mas de un año (duración para realizar la tesis) desarrollar por mi cuenta una página que trabaje por completo con AJAX, aunque en un futuro pienso desarrollar sólo sistemas de este tipo. Otra causa importante es la gran utilización de Javascript, lo que quiero decir es que si por alguna razón el usuario deshabilita de su navegador la ejecución de este lenguaje (lo puede hacer debido a que Javascript se ejecuta en el cliente o computadora del usuario a través de un navegador de internet) solo se mostraría el esqueleto HTML, y obviamente todas las funciones que se hayan detallado con AJAX no tendrían efecto en la página web. Además que el modelo empleado consigue mostrar al usuario secciones que no deben estarse actualizando durante su uso, lo que hace alusión a tener secciones de tipo AJAX empleando código más simple, los Frames.

---

<sup>27</sup> Ibidem p.397

## **CAPÍTULO 3**

# **ANÁLISIS DEL SISTEMA DE ADMINISTRACIÓN Y CONTEO PARA PRESTADORES DE SERVICIO SOCIAL EN LA FES ARAGÓN**

---

---

El alcance de esta investigación tiene como fin establecer una unión entre el estudiante y las nuevas tecnologías, así como fortalecer su relación con los sistemas computacionales, de tal forma que adquiriera el conocimiento suficiente para comprender que las técnicas desarrolladas para la automatización de procedimientos, como las empleadas para el SICSS, son totalmente funcionales, legibles, y sobre todo que minimizan el trabajo y tiempo tanto de los asesores como de los usuarios, y al mismo tiempo se tiene un sistema capaz de almacenar, administrar y controlar a toda una infinidad de alumnos que trascenderán por las próximas generaciones sin pérdida alguna.

A éste tipo de estudio se le denomina como correlacional-explicativo. Básicamente se enfoca en la resolución de inferencias tales como el ¿Por qué de la necesidad por relacionar a los estudiantes con un sistema de cómputo?; ¿Qué tan necesario es seguir implementando sistemas que hagan las cosas por nosotros?; ¿Cómo confiar en los datos que almacenan las computadoras? y ¿Por qué sustituir a los métodos comunes para administrar a los prestadores de servicio social?

Las respuestas son muy simples, ya que los resultados son muy notorios y saltan a la vista en la época actual, en primera instancia tenemos que la computación ya no es un lujo como cuando nació, ni tampoco es un ámbito del que alguien se libraría sólo porque



sus estudios son diferentes a la computación; es tan sencillo como mirar a tu alrededor y apreciar cuan avanzada hace ésta tecnología a cualquier estudio en el mundo, desde libros virtuales que puedes leer en tu celular, hasta sistemas computacionales que han diseñado infraestructuras en países de primer mundo; los alcances tienden a infinito ya que día a día los profesionales nos dedicamos a crear mejores y más innovadores sistemas.

De ahí se desprende la verdadera necesidad del humano por la máquina, pero esto no es una ley, no es necesario que una persona deba tener una computadora o deba utilizar sistemas de cómputo para seguir viviendo, sin embargo si alguien decidiera trabajar con técnicas obsoletas, o analógicamente, los resultados de sus operaciones no serían los mas aceptados ante la actual sociedad, por ejemplo: realizar tu currículum a mano o con una máquina de escribir para entrar en una empresa, o del mismo modo hacer tus trabajos y proyectos de la escuela; administrar tu negocio mediante cuentas mentales y plasmarlas en una libreta; o controlar a una gran cantidad de estudiantes de Universidad que desean hacer su servicio social dentro del plantel, anotándolos en un cuadernillo y sumando sus horas laboradas con una calculadora.

Está muy claro que la dependencia hombre-máquina trascenderá por muchas generaciones, y día a día se seguirán implementando nuevos sistemas que hagan el trabajo pesado por nosotros, minimizándonos el tiempo de realización y con el propósito final de hacernos la vida más fácil.

Al implementar el sistema en la Facultad de Estudios Superiores Aragón se conseguirá:

- Automatizar el proceso de control de tiempo para los prestadores de servicio social, en la Facultad de Estudios Superiores Aragón.
- Se eliminará por completo las técnicas arcaicas para el conteo de horas de los prestadores de servicio.
- La facilidad de registro y consulta de datos se hará evidente tanto para prestadores como para asesores.
- El impacto del sistema será más exitoso que los actualmente utilizados (sin difusión alguna), además de ser más veloz y eficaz.
- La manejabilidad del sistema será tan sencilla como ingresar a tu correo electrónico.
- Con la utilización de este sistema los jefes del área no se preocuparan por ninguna alteración en los datos del tiempo laborado ya que la seguridad del sistema está desarrollada con formas de programación avanzadas que involucran reportes de todos los movimientos y respaldo múltiple de la base de datos.
- La información almacenada podrá servir como comprobante de labor para futuras consultas.

## 3.1 UML Y CASOS DE USO

El Lenguaje Unificado de Modelado (UML) es, como su nombre lo indica, un lenguaje para modelar un sistema de software, está compuesto por una notación muy específica y por reglas semánticas relacionadas para la construcción de los mismos.

Además soporta un conjunto basto de elementos de notación gráfica por lo que el UML en la actualidad es una herramienta muy utilizada por los ingenieros de software.

Para lograr un análisis minucioso del SICSS se requiere aplicar una importante técnica del Leguaje Unificado de Modelado (UML) conocida por “casos de uso”.

*Un caso de uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo<sup>28</sup>, en donde los principales elementos para llevar a cabo el estudio de cada una de las interacciones posibles con el sistema, son los usuarios.*

Es muy importante contemplar todas las opciones posibles para cada caso de uso ya que se trata de un proyecto iterativo, desde la rutina más común o normal hasta la menos esperada, un caso inusual.

Para conseguir éste tipo de información y sin duda a la fuga de algún caso de uso, es apropiado hablar con los usuarios habituales y analizar con ellos las distintas cosas que desean que el sistema haga. De éste modo se logrará detallar cada proceso y las variaciones que éste pueda tener.

Iniciando el análisis me encuentro con una dicotomía en los casos de uso, y es que tienden a tener doble significado, es decir, que se puede confundir las funciones que realiza el sistema con las necesidades del usuario; por ejemplo: “iniciar sesión del usuario”, por un lado tenemos la función del lenguaje de programación PHP para almacenar los datos de entrada en la base de datos, misma que será usada iterativamente para cada prestador de servicio social, pero por otro lado encontramos la necesidad del usuario que es darse de alta en la página web para comenzar a contar su tiempo.

Para resolver esta diferencia se hará una separación de los conceptos, las necesidades del usuario se considerarán como los objetivos, mientras que las funciones del sistema serán las diversas soluciones para tales objetivos, lo que significa que cada objetivo determinara los requisitos funcionales del sistema.

Cada caso de uso se detallará textualmente para que al final del apartado se pueda mostrar de forma visual el conjunto de éstos mediante un diagrama.

---

<sup>28</sup> Fowler Martin, “UML Gota a Gota”, pp. 49

Para detallar cada caso de uso se emplearán los siguientes conceptos:

- **Actor:** Usuario o sistema externo.
- **Precondiciones:** Conjunto de condiciones que deben ser ciertas antes de iniciar el caso de uso.
- **Camino básico:** Es la rutina común del caso de uso. Los pasos a seguir para determinar una postcondición.
- **Variaciones:** Son las variables que contiene el proceso de caso de uso que pueden afectar al resultado.
- **Postcondiciones:** Indican el estado final del proceso después de que el caso de uso termine exitosamente. No es un proceso, sino el resultado.

**Listado clasificado de casos de uso:**

- ✓ *Administradores*
  - Cargar página SICSS
  - Registrar asesor
  - Registrar usuario
  - Usuarios en línea
  - Asesores registrados
  - Modificar datos de asesor
  - Modificar datos de usuario
  - Modificar horas de usuario
  - Modificar lugar
  - Modificar IP
  - Configuración de IP
- ✓ *Asesores*
  - Cargar página SICSS
  - Inicio de sesión de asesor
  - Registrar usuario
  - Usuarios en línea
  - Forzar baja
  - Historial de uso de usuarios
  - Usuarios registrados
- ✓ *Prestadores de servicio social (usuario)*
  - Cargar página SICSS
  - Iniciar conteo
  - Terminar conteo
  - Usuarios en línea
  - Historial de uso de usuarios

- Usuarios registrados
- ✓ *Invitados*
  - Usuarios en línea
  - Usuarios registrados
- ✓ *Servidor de base de datos PostgreSQL*
  - Autenticación de registro de asesor
  - Autenticación de registro de usuario
  - Autenticación de inicio de sesión de asesor
  - Autenticación de inicio de conteo de usuario
  - Autenticación de término de conteo de usuario

### **Casos de uso:**

#### **Caso de uso “Cargar página SICSS”**

Actor: Administrador, asesor, usuario o invitado.

Precondiciones:

- ✓ Tener servicio de internet en la computadora donde se requiere cargar el SICSS.
- ✓ Tener instalado un navegador de internet compatible, tal como: Mozilla Firefox (3.0 ó superior), Internet Explorer (7.0 ó superior), Netscape (9 ó superior), Safari (5 ó superior) ó Google Chrome (4.0 ó superior).

Camino básico:

1. Ejecutar su navegador de internet.
2. Introducir en la barra de direcciones la url: <http://132.248.221.198/sicss> y dar clic en “Ir a” o presionar “Entrar” del teclado.

Variaciones: Al realizar el paso dos del camino básico, existe la posibilidad de que la página no cargue correctamente debido a:

- El servicio de internet se ha caído o dejado de transmitir.
- El servidor 132.248.221.198, portador del SICSS, no funciona correctamente. (Si el servidor deja de funcionar solo tardará unos segundos en que entre en funcionamiento un respaldo del mismo, es decir, otro servidor con características idénticas).
- Algún virus en la computadora donde se está cargando la página ha inhibido el acceso a la red (se puede optar por usar otra computadora, solo si la restricción de IP ha sido deshabilitada, o bien utilizar un programa antivirus para evitar esta variación).

Postcondiciones: Si la computadora, el servidor e internet no tienen conflictos la página SICSS se mostrará en pantalla. En caso contrario el navegador mostrará un error.

### **Caso de uso “Registrar asesor”**

Actor: Administrador.

Precondiciones:

- ✓ Haberse cargado la página SICSS correctamente.
- ✓ Conocer contraseñas administrativas.
- ✓ Tener datos correctos del asesor:
  - Nombre completo.
  - Departamento o lugar donde se hará uso el SICSS.
  - Pseudónimo ó nombre corto.
  - Contraseña.
  - Direcciones IP (opcional)

Camino básico:

1. Dar clic en el botón “Administrar” del menú de botones izquierdo.
2. Llenar los campos “Admin” y “Password” con las contraseñas administrativas proporcionadas al administrador, y dar clic en “ingresar”.
3. Desplazarse con el mouse a “Registrar” en el nuevo menú que apareció en la parte superior y dar clic en la opción “Asesor”.
4. Llenar campos de nombre, apellido paterno, apellido materno, lugar, pseudónimo y contraseña, con los datos correctos del asesor.
5. Llenar campos de direcciones IP (sólo si se requiere usar reconocimiento de computadoras a través de la dirección IP) y dar clic en enviar.
6. Dar clic “salir” para cerrar la sesión del administrador.

Variaciones:

Al realizar el paso dos existe la posibilidad de que la página muestre un error debido a:

- Alguna de las contraseñas se escribió incorrectamente.
- Se dejó algún campo en blanco.

Al realizar el paso cinco existe la posibilidad de que la página muestre un error debido a:

- Ya existe un asesor con el mismo pseudónimo.
- La contraseña no coincide en ambos campos.
- Se dejó algún campo en blanco.
- La dirección IP no es válida o está incompleta.

La dirección IP ya existe.

Estas variaciones dependerán del caso de uso “Autenticación de registro de asesor”.

Postcondiciones: Si los datos fueron enviados correctamente se mostrará un mensaje de confirmación y se podrá hacer uso inmediato del nuevo asesor en el sistema, de lo contrario se mostrará un mensaje de error.

### **Caso de uso “Autenticación de registro de asesor”**

Actor: PostgreSQL.

Precondiciones:

- ✓ No haber dejado vacío ninguno de los campos:
  - Nombre.
  - Apellido paterno.
  - Apellido materno.
  - Lugar.
  - Pseudónimo.
  - Contraseña.
  - Confirmar contraseña.
- ✓ Los campos “contraseña” y “confirmar contraseña” deben ser idénticos.
- ✓ Si se llena alguna dirección IP ésta debe tener 4 segmentos de entre los valores 0 y 255 cada uno.

Camino básico:

1. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente o insertar los nuevos datos.
2. Comprobar que el asesor no esté registrado en la base de datos a través del nombre completo e insertar datos.
3. Comprobar que el pseudónimo esté disponible en la base de datos e insertar datos.
4. Verificar que los campos de dirección IP estén vacíos, de lo contrario analizar si el formato introducido es correcto e insertar nueva dirección en la base de datos.

Variaciones:

En el paso uno del camino básico, es posible que no pueda continuar la autenticación debido a:

La base de datos tuvo algún error interno.

El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

En el paso dos es posible que el asesor ya se haya registrado con anterioridad, de ser así el SICSS enviará un informe para hacer tal indicación, de lo contrario se quedará un aviso interno indicando que este campo está listo para almacenarse.

En el paso tres si el pseudónimo utilizado ya existe en la base de datos se enviará un mensaje de error para que sea cambiado, de lo contrario se quedará un aviso interno indicando que este campo está listo para almacenarse.

Postcondiciones:

Si no hubo errores durante la autenticación se quedarán avisos internos para todos los campos indicando que son correctos y están listos para almacenarse.

### **Caso de uso “Inicio de sesión de asesor”**

Actor: Asesor

Precondiciones:

- ✓ Haberse cargado la página SICSS correctamente.
- ✓ El asesor debe estar registrado en el sistema.

Camino básico:

1. Dar clic sobre el botón azul etiquetado como “Iniciar sesión” de la columna izquierda en la página.
2. Llenar los campos: pseudónimo y contraseña.
3. Presionar “enter” o dar clic en el botón de “iniciar sesión” que se encuentra debajo de los campos para enviar los datos para ser autenticados.

Variaciones:

En el paso uno si no es la primera vez que se inicia sesión es muy probable que ya estén depositados los cookies en la computadora donde se está abriendo la página SICSS, por lo que se hará automáticamente la autenticación para pasar directamente a la página de inicio restringida.

En el paso tres si no se llena alguno de los campos, se enviará un mensaje de error y no se podrá continuar hasta que sean llenados.

Postcondiciones:

De ser correcto el procedimiento se dependerá del módulo “Autenticación de inicio de sesión de asesor” para poder o no ver la página de inicio restringida.

### **Caso de uso “Autenticación de inicio de sesión de asesor”**

Actor: PostreSQL.

Precondiciones:

- ✓ El asesor debe estar registrado en el sistema.
- ✓ No haber dejado vacío ninguno de los campos:
  - Asesor
  - Contraseña
- ✓ Presionar el botón “iniciar sesión” después de llenar los campos mencionados en la precondición anterior.

Camino básico:

1. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente.
2. Comprobar que el asesor esté registrado en la base de datos, a través del pseudónimo.
3. Comprobar que la contraseña coincida con la registrada asociada al pseudónimo en la base de datos.
4. Depositar archivos (cookies) en la computadora del asesor que contengan la información de su inicio de sesión para la no repetir el proceso de autenticación en próximas ocasiones.

Variaciones:

En el paso uno del camino básico, es posible que no pueda continuar la autenticación debido a:

- La base de datos tuvo algún error interno.
- El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

En el paso dos es posible que el asesor no esté registrado, de ser así el SICSS enviará un informe para hacer tal indicación, de lo contrario se podrá continuar la autenticación.

En el paso tres si la contraseña introducida no coincide con la asociada al pseudónimo en la base de datos, se enviará un informe para hacer tal indicación, de lo contrario se habrá completado satisfactoriamente la comprobación de identidad.



En el paso cuatro es posible que no se puedan depositar los cookies si es que el navegador de internet tiene deshabilitada la recepción de cookies, lo cual no afecta solo se tendrá que repetir el proceso de inicio de sesión siempre que se vuelva a cargar la página.

Postcondiciones:

Si no hubo errores durante la autenticación se dará paso a la página de inicio restringida, en donde los prestadores de servicio social podrán comenzar a hacer uso de ella.

### **Caso de uso “Registrar usuario”**

Actor: Administrador o asesor.

Precondiciones:

- ✓ El asesor debe haber iniciado sesión.
- ✓ Tener datos correctos del usuario:
  - Nombre completo.
  - Número de cuenta
  - Departamento o lugar donde se hará el servicio social.
  - Nombre de usuario (nickname).
  - Contraseña
  - Lugar (departamento donde se laborará)
  - Licenciatura.

Camino básico para administradores:

1. Dar clic en el botón “Administrar” del menú de botones izquierdo.
2. Llenar los campos “Admin” y “Password” con las contraseñas administrativas proporcionadas al administrador, y dar clic en “ingresar”.
3. Desplazarse al menú desplegable “Registrar” ubicado en la parte superior izquierda y dar clic en la opción “Usuario”.
4. Llenar los campos mencionados en el punto tres de las precondiciones y dar clic en enviar.

Camino básico para asesores:

1. Dar clic en el botón “Iniciar sesión” del menú de botones izquierdo.
2. Llenar los campos “asesor” y “contraseña” con los datos del asesor y dar clic en “iniciar sesión”.
3. Desplazarse al menú desplegable “Asesor” ubicado en la parte superior izquierda y dar clic en la opción “Registrar usuario”.

4. Llenar los campos mencionados en el punto tres de las precondiciones y dar clic en enviar.

Variaciones:

Al realizar el paso dos de los caminos básicos existe la posibilidad de que la página muestre un error debido a:

- . Alguna de los campos se escribió con datos incorrectos.
- . Se dejó algún campo en blanco.

Al realizar el paso cuatro existe la posibilidad de que la página muestre un error debido a:

- . Se ha registrado el usuario con anterioridad.
- . Ya existe un prestador de servicio social con el mismo nombre de usuario.
- . La contraseña no coincide en ambos campos.
- . Se dejó algún campo en blanco.

Postcondiciones: Si los datos fueron enviados correctamente se mostrará un mensaje de confirmación y se podrá hacer uso inmediato del nuevo usuario en el sistema, de lo contrario se mostrará un mensaje de error.

### **Caso de uso “Autenticación de registro de usuario”**

Actor: PostgreSQL.

Precondiciones:

- ✓ No haber dejado vacío ninguno de los campos:
  - . Número de cuenta.
  - . Nombre.
  - . Apellido paterno.
  - . Apellido materno.
  - . Carrera.
  - . Nombre de usuario.
  - . Contraseña.
  - . Confirmar contraseña.
  - . Lugar
- ✓ Los campos “contraseña” y “confirmar contraseña” deben ser idénticos.
- ✓ Presionar el botón “enviar” después de llenar los campos mencionados en la segunda precondición.

Camino básico:

1. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente.
2. Comprobar que el usuario no esté registrado en la base de datos, a través del nombre y apellidos.
3. Comprobar que el nombre de usuario esté disponible en la base de datos.
4. Verificar que el lugar corresponda al registrado con el asesor que tiene la sesión activa.

Variaciones:

En el paso uno del camino básico, es posible que no pueda continuar la autenticación debido a:

- . La base de datos tuvo algún error interno.
- . El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

En el paso dos es posible que el usuario ya se haya registrado con anterioridad, de ser así el SICSS enviará un informe para hacer tal indicación, de lo contrario se quedará un aviso interno indicando que este campo está listo para almacenarse.

En el paso tres si el nombre de usuario utilizado ya existe en la base de datos se enviará un mensaje de error para que sea cambiado, de lo contrario se quedará un aviso interno indicando que este campo está listo para almacenarse.

El paso cuatro aplica cuando un asesor registra un usuario. Si el lugar difiere del registrado con la sesión del asesor activo, se enviará un mensaje de error pidiendo la corrección del mismo.

Postcondiciones:

Si no hubo errores durante la autenticación se dará autorización al servidor de base de datos PostgreSQL para almacenar la información.

### **Caso de uso “Iniciar conteo”**

Actor: Usuario.

**Precondiciones:**

- ✓ El asesor debe haber iniciado sesión.

**Camino básico:**

1. Deslizar el mouse hacia el menú desplegable “inicio”, dar clic sobre la opción “Iniciar conteo”.
2. Llenar los campos: nombre de usuario y contraseña.
3. Presionar “enter” o dar clic en el botón de “iniciar” que se encuentra debajo de los campos para enviar los datos y ser autenticados.

**Variaciones:**

En el paso tres si no se llena alguno de los campos, se enviará un mensaje de error y no se podrá continuar mientras alguno esté vacío.

**Postcondiciones:**

De ser correcto el procedimiento se dependerá del módulo “Autenticación de inicio de sesión de usuario” para poder ver o no la página de inicio restringida.

**Caso de uso “Autenticación de inicio de conteo de usuario”**

Actor: PostgreSQL.

**Precondiciones:**

- ✓ No haber dejado vacío ninguno de los campos:
  - Nombre de usuario
  - Contraseña
- ✓ Presionar el botón “iniciar sesión” después de llenar los campos mencionados en la precondición anterior.

**Camino básico:**

1. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente.
2. Comprobar que el usuario esté registrado en la base de datos, a través del nombre de usuario.
3. Comprobar que la contraseña coincida con la registrada asociada al nombre de usuario en la base de datos.
4. Comprobar que el usuario no tenga iniciado ya su conteo de tiempo.

5. Preparar la siguiente información para que PostgreSQL la almacene en la tabla “sesiones”.

- ✓ Número de cuenta
- ✓ Fecha de entrada
- ✓ Hora de entrada

Variaciones:

En el paso uno del camino básico, es posible que no pueda continuar la autenticación debido a:

- . La base de datos tuvo algún error interno.
- . El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

En el paso dos es posible que el usuario no esté registrado, de ser así el SICSS enviará un informe para hacer tal indicación, de lo contrario se podrá continuar la autenticación.

En el paso tres si la contraseña introducida no coincide con la asociada al nombre de usuario en la base de datos se enviará un informe para hacer tal indicación, de lo contrario se habrá completado satisfactoriamente la autenticación.

En el paso cuatro si el usuario ya tiene su sesión activa se enviará un mensaje con la información de su inicio de sesión (fecha y la hora de entrada).

Postcondiciones:

Si no hubo errores durante la autenticación se dará autorización al servidor de base de datos PostgreSQL para almacenar temporalmente los datos del paso 5 del camino básico.

### **Caso de uso “Usuarios en línea”**

Actor: Administrador, asesor, usuario o invitado.

Precondiciones:

- ✓ El asesor de área debe haber iniciado sesión en el sistema, o el administrador en la sección de “Administrar”.

Camino básico:

1. Deslizarse al menú desplegable “Consultar” y elegir la opción “usuarios en línea”

2. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente y se selecciona únicamente a los usuarios que estén en la tabla “sesiones” para mostrarlos en pantalla.

Variaciones:

En el paso dos del camino básico, es posible que no pueda mostrar ninguna información debido a:

- . Ningún usuario ha comenzado su conteo en el sistema.
- . La base de datos tuvo algún error interno.
- . El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

Postcondiciones:

Si no hay errores en el procedimiento la página debe mostrar una lista con todos los usuarios que tengan su conteo de tiempo activo y la información de cada uno de ellos.

### **Caso de uso “Terminar conteo”**

Actor: Usuario.

Precondiciones:

- ✓ El usuario debe haber iniciado su conteo de tiempo

Camino básico:

1. Deslizarse al menú desplegable “Inicio” y elegir la opción “Terminar conteo”.
2. Llenar los campos: nombre de usuario y contraseña.
3. Presionar “enter” o dar clic en el botón de “Terminar” que se encuentra debajo de los campos para enviar los datos y ser autenticados.

Variaciones:

En el paso tres si no se llena alguno de los campos, se enviará un mensaje de error y no se podrá continuar mientras alguno esté vacío.

Postcondiciones:

De ser correcto el procedimiento se dependerá del módulo “Autenticación de término de conteo de usuario” para poder ver o no la información de término de conteo de tiempo.

## Caso de uso “Autenticación de término de conteo de usuario”

Actor: PostgreSQL.

Precondiciones:

- ✓ No haber dejado vacío ninguno de los campos:
  - Nombre de usuario
  - Contraseña
- ✓ Presionar el botón “Terminar” después de llenar los campos mencionados en la precondición anterior.

Camino básico:

1. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente.
2. Comprobar que el usuario esté registrado en la base de datos, a través del nombre de usuario.
3. Comprobar que la contraseña coincida con la registrada asociada al nombre de usuario en la base de datos.
4. Comprobar que el usuario haya iniciado su conteo de tiempo.
5. Preparar la siguiente información para que PostgreSQL la almacene en las tablas “historiales” y “passwords”.
  - ✓ Número de cuenta
  - ✓ Hora de entrada
  - ✓ Hora salida
  - ✓ Horas del día
  - ✓ Total de horas
6. Mostrar datos informativos al usuario.

Variaciones:

En el paso uno del camino básico, es posible que no pueda continuar la autenticación debido a:

- La base de datos tuvo algún error interno.
- El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

En el paso dos es posible que el usuario no esté registrado, de ser así el SICSS enviará un informe para hacer tal indicación, de lo contrario se podrá continuar la autenticación.

En el paso tres si la contraseña introducida no coincide con la asociada al nombre de usuario en la base de datos se enviará un informe para hacer tal indicación, de lo contrario se habrá completado satisfactoriamente la autenticación.

En el paso cuatro si el usuario no ha iniciado su conteo de tiempo se enviará un mensaje indicando que el usuario no puede terminar su conteo ya que no lo ha iniciado.

En el paso seis el mensaje de información mostrará el tiempo que duró la sesión del usuario y su tiempo total acumulado, solo en caso que se hayan cumplido las 480 horas reglamentarias se enviará un mensaje de felicitación y el usuario no podrá seguir acumulando tiempo.

Postcondiciones:

Si no hubo errores durante la autenticación se dará autorización al servidor de base de datos PostgreSQL para almacenar los datos del paso 5 del camino básico.

### **Caso de uso “Forzar baja”**

Actor: Asesor.

Precondiciones:

- ✓ El asesor debe haber iniciado sesión.

Camino básico:

1. Deslizarse al menú desplegable “Asesor” y elegir la opción “Forzar baja”.
2. Introducir contraseña del asesor.
3. Ubicar al usuario que se desea dar de baja del sistema y dar clic al botón “Baja”.

Variaciones:

En el paso dos si no se escribe la contraseña correctamente en tres ocasiones quedará bloqueada la opción por el resto del día.

Postcondiciones:

Al finalizar el camino básico se verá la información mostrada por el caso de uso “Autenticación de término de conteo de usuario” como si hubiese sido exitoso su procedimiento.

### **Caso de uso “Historial de uso de usuarios”**

Actor: Administrador, asesor o usuario.



### Precondiciones:

- ✓ El asesor de área debe haber iniciado sesión en el sistema, o el administrador en la sección de “Administrar”.

### Camino básico:

1. Deslizarse al menú desplegable “Consultar” y elegir la opción “Historial”
2. Se hace una conexión con la base de datos PostgreSQL para hacer consultas a los datos registrados anteriormente y se seleccionan todos los registros que estén en la tabla “historiales” para mostrarlos en pantalla ordenadamente.
3. Escribir en el área de “Mostrar solo resultados de” el número de cuenta del usuario que se desea consultar y dar clic en “Buscar”.

### Variaciones:

En el paso dos del camino básico es posible que no pueda mostrar ninguna información debido a:

- . Ningún usuario ha utilizado el sistema aún.
- . La base de datos tuvo algún error interno.
- . El servidor que contiene la base de datos perdió la conexión por falta de energía eléctrica, o la falla de algún hardware.

El paso tres del camino básico no es completamente necesario de utilizar, sin embargo al hacerlo se consigue mostrar únicamente las entradas del usuario elegido.

### Postcondiciones:

Si no hay errores en el procedimiento la página debe mostrar una lista con todas las entradas de todos los usuarios que han utilizado el SICSS en ese lugar.

## **Caso de uso “Cierre de sesión de asesor”**

Actor: Asesor

### Precondiciones:

- ✓ El asesor debe haber iniciado su sesión en el sistema.

### Camino básico:

1. Dar clic sobre el botón azul etiquetado como “Cerrar sesión” posicionado en la parte superior derecha de la página.

Variaciones:

Al dar clic al botón cerrar sesión, no habrá lugar para preguntas al usuario, por lo que la sesión será cerrada de inmediato.

Postcondiciones:

Después de dar clic se eliminarán los registros de sesión del asesor, pero las personas que tengan activos sus conteos en el sistema no sufrirán cambios.

### **Caso de uso “Asesores registrados”**

Actor: Administrador.

Precondiciones:

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.

Camino básico:

1. Dar clic en el menú desplegable “Consultar → Asesores”

Variaciones:

En caso de no existir asesores registrados se mostrará un mensaje con tal indicación.

Postcondiciones:

Después entrar al menú se desplegará una lista con todos los asesores registrados y su respectiva información, además de mostrar si éste se encuentra actualmente en con su sesión activa o no.

### **Caso de uso “Usuarios registrados”**

Actor: Administrador, asesor, usuario o invitado.

Precondiciones:

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas, o el asesor tener su sesión activa.

Camino básico:

### 1. Dar clic en el menú desplegable “Consultar → Usuarios registrados”

Variaciones:

En caso de no existir usuarios registrados se mostrará un mensaje con tal indicación.

Postcondiciones:

Después entrar al menú se desplegará una lista con todos los usuarios registrados, además de tener un pequeño buscador donde por medio del número de cuenta se puede ver la información detallada de un solo usuario, para el caso de los asesores se mostrarán sólo los usuarios de su lugar correspondiente.

### Caso de uso “Modificar datos de asesor”

Actor: Administrador.

Precondiciones:

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.
- ✓ Debe existir por lo menos un asesor registrado.

Camino básico:

1. Dar clic en el menú desplegable “Modificar → Datos de asesor”.
2. Elegir la opción “Editar” situada a un costado del registro del asesor (en la misma fila) para cambiar alguno de sus datos, o elegir “Borrar” para eliminar el registro.
3. Editar los datos correspondientes y dar clic al botón “Guardar cambios” situado al final del formulario, o bien dar clic en “Regresar” para elegir otra opción o asesor.

Variaciones:

En el paso dos del camino básico se puede emplear el atajo “Control + F” del navegador para buscar algún registro en caso de ser bastantes. Si se elige la opción “Borrar” se enviará un mensaje de alerta indicando que la acción es irreversible y puede afectar a uno o varios usuarios del sistema.

En el paso tres se pueden llenar opcionalmente los campos de contraseña, en caso de introducir la original erróneamente, se actualizarán los demás datos del formulario excepto esos; si se escribe correctamente la contraseña original y las dos nuevas son idénticas, se actualizará el registro de seguridad correctamente, pero si se dejan vacíos los campos sencillamente se omitirá esa sección.

Postcondiciones:

De ser correcto el procedimiento en esta sección se obtendrá un registro de asesor actualizado, o bien la supresión del mismo.

### **Caso de uso “Modificar datos de usuario”**

Actor: Administrador.

Precondiciones:

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.
- ✓ Debe existir por lo menos un asesor y un usuario registrado.

Camino básico:

1. Dar clic en el menú desplegable “Modificar → Datos de usuario”.
2. Elegir la opción “Editar” situada a un costado del registro del usuario (en la misma fila) para cambiar alguno de sus datos, o elegir “Borrar” para eliminar el registro.
3. Editar los datos correspondientes y dar clic al botón “Guardar cambios” situado al final del formulario, o bien dar clic en “Regresar” para elegir otra opción o asesor.

Variaciones:

En el paso dos del camino básico se puede emplear el atajo “Control + F” del navegador para buscar algún registro en caso de ser bastantes. Si se elige la opción “Borrar” se enviará un mensaje de alerta preguntando al administrador si está seguro de realizar la acción.

En el paso tres se pueden llenar opcionalmente los campos de contraseña, en caso de introducir la original erróneamente, se actualizarán los demás datos del formulario excepto esos; si se escribe correctamente la contraseña original y las dos nuevas son idénticas, se actualizará el registro de seguridad correctamente, pero si se dejan vacíos los campos sencillamente se omitirá esa sección.

Postcondiciones:

De ser correcto el procedimiento en esta sección se obtendrá un registro de usuario actualizado, o bien la supresión del mismo.

### **Caso de uso “Modificar horas de usuario”**

Actor: Administrador.

**Precondiciones:**

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.
- ✓ Debe existir por lo menos un asesor y un usuario registrado.

**Camino básico:**

1. Dar clic en el menú desplegable “Modificar → Horas de usuario”.
2. Elegir la opción “Editar” situada a un costado del registro del usuario (en la misma fila) para agregar o quitar horas de labor, o escribir el número de cuenta del usuario a modificar en el buscador que se sitúa a un costado derecho de la página.
3. Editar los datos correspondientes y dar clic al botón “Sumar” para agregar el tiempo escrito, o en “Restar” para quitárselo, o bien dar clic en “Regresar” para elegir otro usuario.

**Variaciones:**

En el paso dos del camino básico es posible que se muestre el mensaje “No se han encontrado registros”, debido a que se escribió un número de cuenta inexistente en el buscador.

**Postcondiciones:**

De ser correcto el procedimiento en esta sección se obtendrá un registro de horas laboradas por el usuario actualizado, y se registrará en su historial tal movimiento.

**Caso de uso “Modificar lugar”**

Actor: Administrador.

**Precondiciones:**

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.

**Camino básico:**

1. Dar clic en el menú desplegable “Modificar → Lugar”.
2. Elegir la opción “Borrar” situada a un costado del registro del usuario (en la misma fila) para eliminar tal registro, o bien dirigirse al final de la página, escribir un nuevo lugar en el campo “Agregar lugar” y dar clic en el botón “Guardar”.

**Variaciones:**

En el paso dos del camino básico es posible que se muestre el mensaje “El lugar está en uso y no puede eliminarse”, en caso que el lugar a eliminar ya esté asignado a algún asesor. Si se intenta agregar un lugar que ya existe no se podrá realizar la acción.

Postcondiciones:

De ser correcto el procedimiento en esta sección se obtendrá un nuevo lugar registrado, o bien la supresión de alguno de ellos.

### **Caso de uso “Modificar IP”**

Actor: Administrador.

Precondiciones:

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.

Camino básico:

1. Dar clic en el menú desplegable “Modificar → IP”.
2. Elegir la opción “Borrar” situada a un costado del registro del usuario (en la misma fila) para eliminar tal registro, o bien dirigirse al final de la página, escribir una nueva dirección IP en el campo “Agregar IP”, elegir un lugar de la lista desplegable en donde se encuentra dicha computadora y dar clic en el botón “Guardar”.

Variaciones:

En el paso dos del camino básico puede no encontrarse el lugar requerido, por lo que tendrá que agregarse desde “Agregar lugar”. Si se intenta agregar una IP que ya existe no se podrá realizar la acción.

Postcondiciones:

De ser correcto el procedimiento en esta sección se obtendrá una nueva dirección IP registrada, o bien la supresión de alguna de ellas.

### **Caso de uso “Configuración de IP”**

Actor: Administrador.

Precondiciones:

- ✓ El administrador debe haber entrado en la sección “Administrar” con las contraseñas administrativas.

Camino básico:

1. Dar clic en el menú desplegable “Configuración → IP”.
2. Presionar el botón “Habilitar” si el sistema de reconocimiento de IP se encuentra Deshabilitado, o “Deshabilitar” si está habilitado.

Variaciones:

Si el sistema está habilitado se restringirá el acceso sólo a las direcciones IP que estén registradas.

Postcondiciones:

De ser correcto el procedimiento en esta sección se verá un texto informativo indicando el estado del sistema de reconocimiento de IP.

## 3.2 DIAGRAMA DE CASOS DE USO

Para conseguir la explicación de los casos de uso de una forma visual se introducirá el concepto de diagrama de casos de uso, que no es más que una representación gráfica de ellos.

En la elaboración de dicho diagrama se requiere nombrar de manera general a todos los tipos de usuario que interactúan con el sistema, es decir los actores, como ya se mencionó éstos pueden ser cualquier cosa externa como un humano u otro sistema, además serán vistos como roles y no como un individuo en particular.

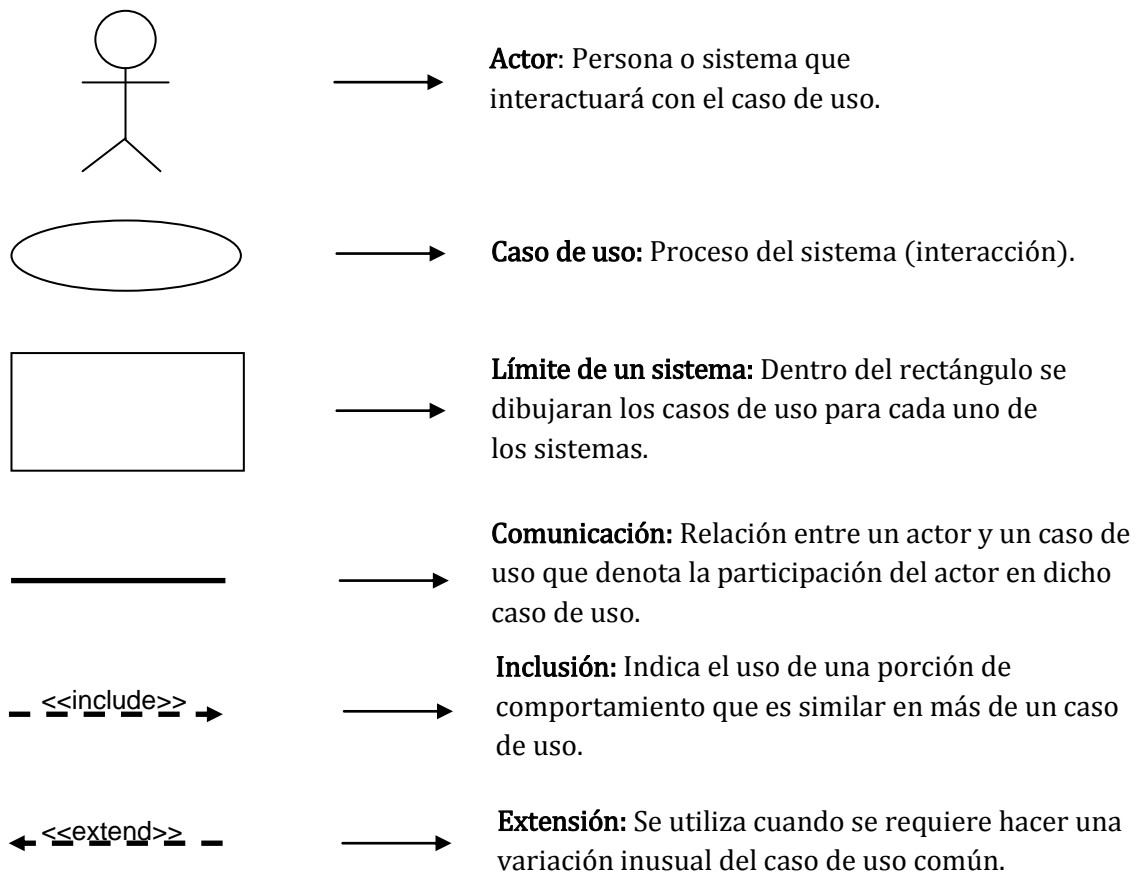
El uso del SICSS contempla a cinco actores: administrador, asesor, prestador de servicio social (usuario), invitado y PostgreSQL (gestor de base de datos):

- ✓ El administrador se encargará de realizar registros de asesores y usuarios entre otras herramientas opcionales del SICSS, así como cualquier modificación al código fuente del sistema, es decir, tiene el control total.
- ✓ El asesor tendrá la anuencia para registrar o dar de baja usuarios (prestadores de servicio social), así como el contacto con los administradores para realizar tareas de modificación y actualización de datos en el sistema.

- ✓ El usuario será el prestador de servicio social, quien utilizará el sistema SICSS para llevar un control de tiempo legítimo durante sus 480 horas de labor.

Cabe resaltar que la diferencia entre ellos no la da el tipo de persona sino el papel que juegan dentro del medio. Por ejemplo, un ingeniero en computación que normalmente sería el administrador puede realizar el papel de un asesor, o viceversa, o hasta puede ser usuario del sistema. Por ello es necesario hacer el análisis tomando los papeles de los actores y no el “título” de la persona.

Otra variable dentro de la elaboración de un diagrama de caso de uso es la forma en que se unirán los actores con cada caso de uso. Se empleará la siguiente simbología para diferenciar las diversas opciones que el sistema contiene:



El diagrama se presenta a continuación:



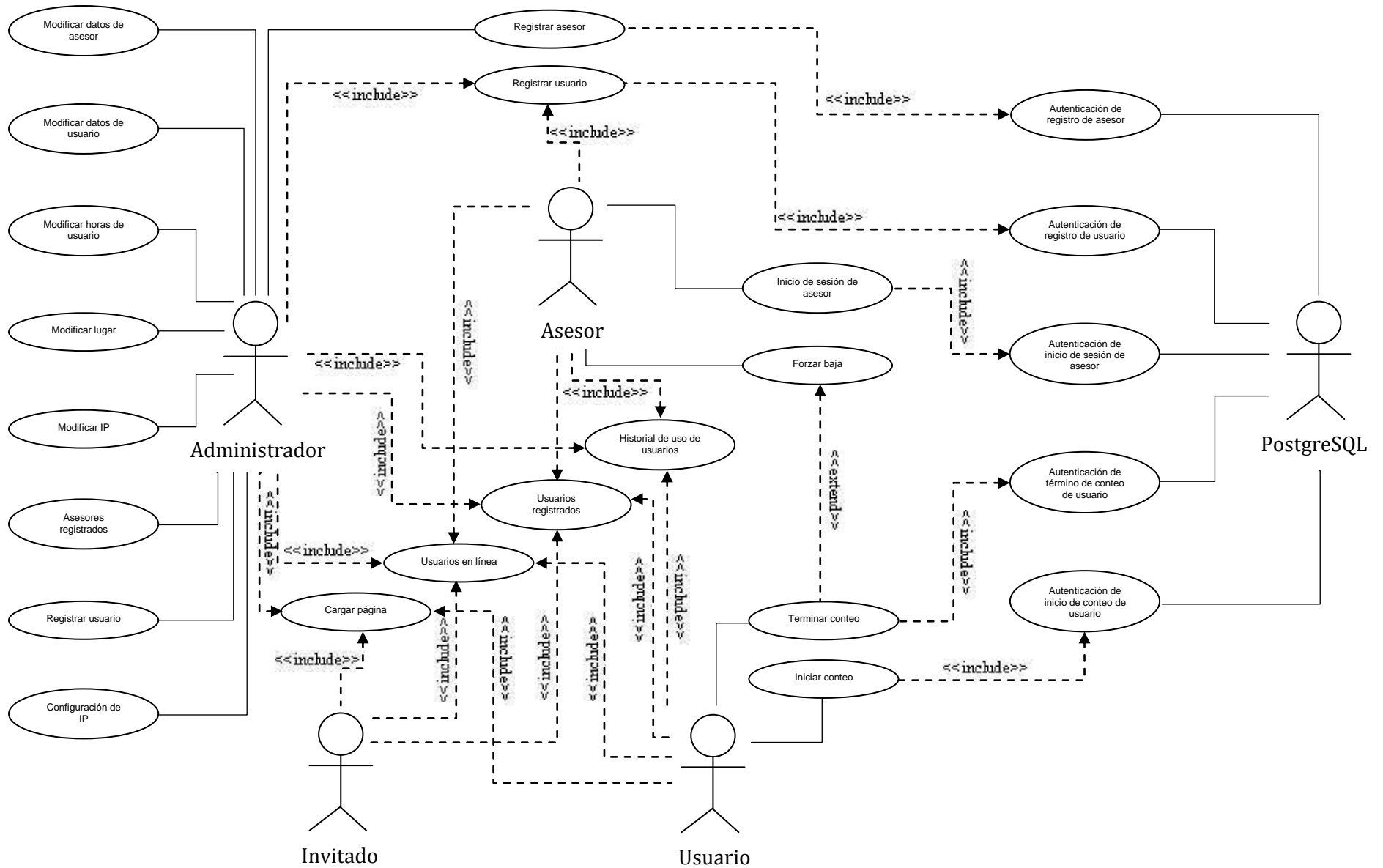


Figura 3.1 – Diagrama de casos de uso del SICSS.

## **CAPÍTULO 4**

# **DISEÑO Y CONSTRUCCIÓN DEL SISTEMA DE ADMINISTRACIÓN Y CONTEO PARA PRESTADORES DE SERVICIO SOCIAL EN LA FES ARAGÓN**

---

---

La manera más eficaz para automatizar el procedimiento de control de horas de los prestadores de servicio social mediante un sistema de cómputo es, desarrollar una interfaz gráfica para facilitar el entendimiento de cualquier usuario, es decir, una página de internet a la cual se pueda entrar mediante un nombre de usuario y contraseña para poder identificarlo y así se inicie automáticamente el conteo de su tiempo; al finalizar su labor del día deberá realizar el mismo procedimiento para indicar al sistema su salida y que éste detenga el conteo.

El sistema tiene la capacidad de almacenar un número bastante grande de usuarios, en la práctica se asegura que ésta no será una limitante porque los servidores empleados para almacenamiento masivo de información en la base de datos, tiene una cabida suficiente para registrar a más del número total de universitarios que habitan la Facultad de Estudios Superiores Aragón.

En los capítulos anteriores ya se ha mostrado toda la fase conceptual, de planificación y análisis para desarrollar una página web, ahora se detallará toda la parte de diseño y construcción de dicho sistema, tal como la utilización de los lenguajes de programación a nivel “código”, es decir, toda la sección operativa como (HTML, PHP, JavaScript, DHML, XML y AJAX), y las aplicaciones para diseño gráfico: Adobe® Flash® y Adobe®

Photoshop®; cabe resaltar que el lenguaje de programación CSS funciona a nivel código (no es aplicación) pero ayudará a mejorar el diseño web.

## 4.1 MARCOS HTML (FRAMES)

Una vez que ya conocemos exactamente lo que debemos hacer hace falta llevarlo a la realidad, lo primero que se va a construir es el esquema o esqueleto que dividirá en tres secciones a la página, mejor conocido como “Frames (marcos)”.

Dicho esquema será programado en HTML ya que su única función es crear tres paneles, dos estáticos y un dinámico, es decir, dos secciones de pantalla que siempre tendrán el mismo contenido (banner y botones), y un dinámico que será el que utilizará la mayor parte de la pantalla porque ahí se mostraran los resultados de cada petición que se haga al sistema, ya sea una pantalla de bienvenida, formularios de registro, submenús y toda la información que gestionará el SICSS.

Durante este capítulo se mostrarán pequeños trozos de código de programación para reforzar cada uno de los detalles que se citan, pero que quede claro que en este proyecto no se trata de explicar cómo funciona cada uno de los lenguajes de programación.

Inicialmente debo hacer énfasis en tres palabras que utilizaré a lo largo de este apartado. *Archivo*: éste no tiene mayor complicación ya que hace referencia a cualquier archivo con la extensión .php, .js, .css, .txt, .html ó .xml. *Módulo*: lo usaré para referenciar a los archivos que contienen procesos del servidor, es decir, los de PHP; y *página*: que hace referencia al código que muestra algo en pantalla.

Básicamente todos los módulos y/o páginas que contendrán la mayoría del código tendrán en común ciertas instrucciones, o mejor dicho, etiquetas HTML. A continuación se muestran cuales son y su respectiva explicación.

1. `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`
2. `<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">`
3. `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`
4. `<head>`
5. `<style media="all" type="text/css"> @import "estilos.css";</style>`
6. `<link rel="shortcut icon" href="img/logo_unam.gif">`
7. `<title>Facultad de Estudios Superiores Arag&oacute;n - SICSS</title>`
8. `</head>`
9. `<body>`

10. `</body>`

11. `</html>`

En el inciso 1 se contiene a la Definición del Tipo de Documento (DTD), para este caso es una página del tipo “Framset” que es de lo que he estado tratando. El inciso 2 y 11 corresponden al inicio y final del contenido HTML, entre estas dos etiquetas se escribirá el contenido de cualquier módulo a realizar, y dentro de la etiqueta del inciso 2 se indica el idioma a utilizar (español).

En el inciso 3 se define el tipo de codificación a utilizar, que para este proyecto se empleará el UTF-8, esta declaración incluye a los símbolos no comunes para que sean vistos por el usuario correctamente, tales como la “ñ” y vocales acentuadas “á”, “é”, “í”, “ó” y “ú”. El inciso 4 y 8 determinan la cabecera de la página, apertura y cierre respectivamente, entre estas dos etiquetas colocaré las llamadas a archivos de estilo CSS (inciso 5) lo que significa la inclusión de otro archivo para ponerle estilos a ese módulo, como alineaciones, tamaños y colores; dentro de la misma cabecera `<head>` se contiene un pequeño icono del logotipo de la UNAM que se mostrará en la pestaña del navegador web (inciso 6) y el título de la página que también se mostrará en la pestaña (inciso 7).

Finalmente en el inciso 9 y 10 se tienen las etiquetas que contendrán entre ellas a todo el cuerpo de ese módulo o página.

Ahora que ya tenemos el código principal no será necesario mostrarlo a lo largo de cada modulo a programar, ya que se tiene entendido que se utilizará en todos los módulos o páginas.

Las etiquetas para formar la estructura de marcos antes mencionada son las siguientes:

1. `<frameset rows='0,120,*' cols='*' border='3'>`
2. `<frame>`
3. `<frame src='Cabecera.html' name='topFrame' scrolling='no' noresize bordercolor='black'>`
4. `<frameset rows='*' cols='134,*' framespacing='0'>`
5. `<frame src='Botones.htm' name='leftFrame' scrolling='auto' noresize bordercolor='black'>`
6. `<frame src='inicio.html' name='mainFrame' scrolling='auto'>`
7. `</frameset>`
8. `</frameset>`
9. `<noframes>`
10. `<body>`
11. `<h1>Tu navegador no soporta la vista con marcos, descarga alguno compatible</h1>`
12. `<h2>Se te recomienda <a href='firefox.exe'>Mozilla Firefox</a> o <a href='http://www.google.com/chrome'>Google Chrome</a></h2>`
13. `</body>`
14. `</noframes>`

De aquí en adelante no explicaré tan minuciosamente el código empleado, sólo las áreas más importantes para conseguir el esclarecimiento completo del diseño, posteriormente mostraré el resultado a través de una imagen.

El código declara inicialmente que existirá una página de marcos en dónde el primero (*top-frame*) será de 120 pixeles verticalmente y horizontalmente todo el espacio, después se indica qué archivo debe colocarse en ese marco, en este caso es "Cabecera.html"; posteriormente se declara otra sección de la página (*left-frame*) que tendrá 134 pixeles de forma horizontal y todo el espacio restante para lo vertical, también se indica el archivo que debe mostrarse en ese marco, "botones.html". Finalmente al espacio restante (*main-frame*), el de mayor tamaño, se le adjunta el archivo de página de inicio, es decir "inicio.html".

En caso que se abra la página con un navegador antiguo que no soporte el contenido con marcos, se mostrará un mensaje de error recomendando al usuario otros navegadores para descargar de internet.

Hasta ahora se ha hecho mención de cuatro archivos de los cuales no se ha desarrollado su contenido: estilos.css, Cabecera.html, Botones.html e inicio.html; es necesario crear estos archivos aunque no tengan ningún contenido, únicamente para probar la funcionalidad del archivo que contiene a los marcos que acabo de explicar al cual nombraré como: "index.php".

Todos los navegadores de internet interpretan por default primero a un solo archivo llamado index.php o index.html, dependiendo del navegador varía cual es el que interpreta primero, en caso que existan estos dos archivos en la misma carpeta.

Entonces, el resultado de index.php se muestra a continuación.

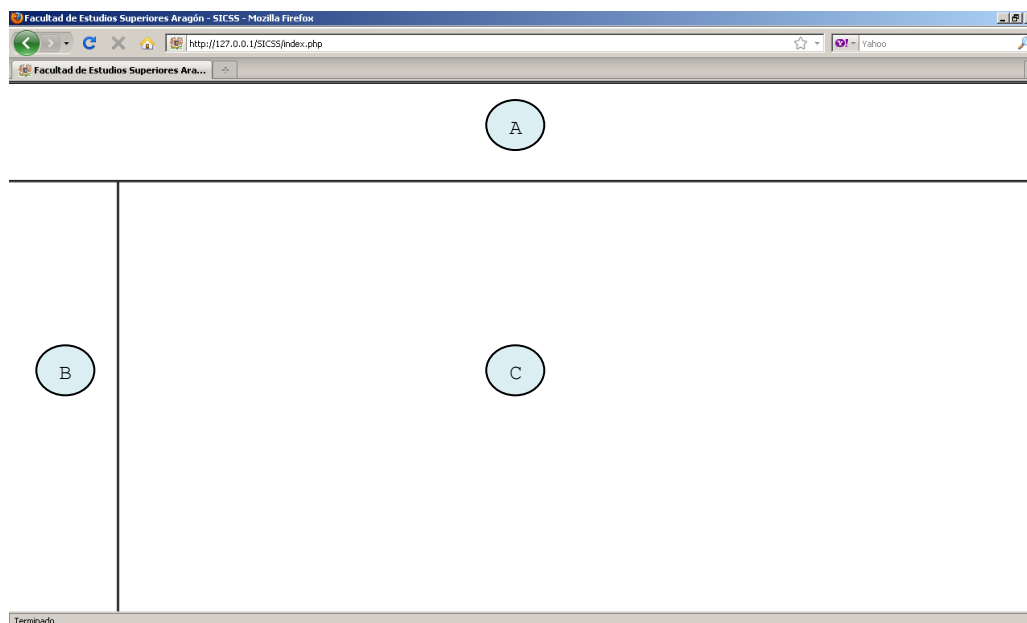


Figura 4.1 - Marcos de la página web.

Donde:

A= Cabecera (top-frame): Lugar fijo de la página que mostrará una animación Flash de presentación

B= Botones (left-frame): Lugar fijo de la página que contendrá 4 botones Flash para ingresar a las diversas opciones del sistema.

C=Contenido (main-frame): Lugar dinámico de la página para mostrar los diversos contenidos en ella.

## 4.2 ANIMACIÓN FLASH (CABECERA TOP-FRAME)

El primer archivo a desarrollar es contiene una cabecera (Cabecera.html), ésta utilizará aproximadamente un 15% de la pantalla (120 pixeles), creada para contener una animación hecha con el software Adobe® Flash® y colocado en el ya mencionado “top-frame”.

Cabe aclarar que previamente me di a la tarea de investigar la resolución a utilizar en las computadoras de la FES Aragón, y se tendrá un estándar de 1024x768 pixeles, basándome en este número es como manejaré los porcentajes.

La elaboración de la animación no se tratará en el proyecto ya que sería muy prolongada su explicación y es un tema que no tiene contexto con la tesis, sencillamente se utilizará como un complemento para darle vista a la página.

El archivo que contiene a la animación está en formato “swf”, dicho elemento será incorporado mediando código HTML de la siguiente manera:

1. `<body background="fondo.jpg">`
2. `<table align="center">`
3. `<td>`
4. `<embed src="banner.swf" quality="high"`  
`pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=Shock`  
`waveFlash" type="application/x-shockwave-flash" width="1000" height="152"`  
`align="center"></embed>`
5. `</td>`
6. `</table>`
7. `</body>`

Aquí sólo tenemos una línea de código importante, la 3, ya que es la encargada de insertar la animación en el archivo Cabecera.html, sólo se hace referencia al archivo banner.swf, que es quien contiene la animación, y después algunos parámetros necesarios para el correcto funcionamiento de la animación sobre internet, finalmente el tamaño que ocupará en la página.

Las líneas 1, 2, 4 y 5 sólo declaran una tabla para conseguir alinear la animación en el centro, el resultado de “Cabecera.html” es el siguiente:

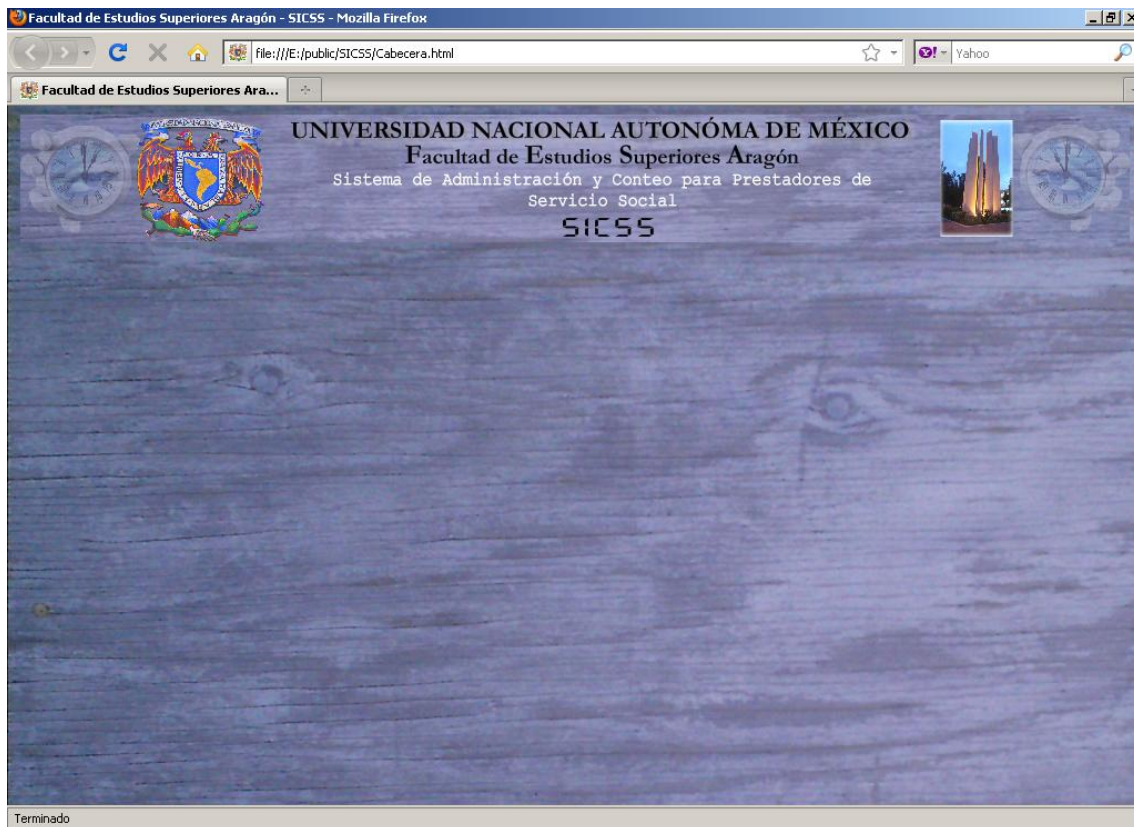


Figura 4.2 - Cabecera de la página web.

### 4.3 MENÚ DE BOTONES FLASH (LEFT-FRAME)

Este archivo ya se sabe que contendrá a 4 botones de tipo Flash (swf) para direccionar a las diversas secciones que están por programarse. Dichos botones son:

- INICIO
- INICIAR SESIÓN
- ADMINISTRAR
- ACERCA DE...

Para conseguir el diseño utilizaremos nuevamente el software Adobe® Flash®. Las características de los botones se muestran a continuación:

#### Botón Inicio

- Texto estático: INICIO.
- Texto dinámico: Ir a página de bienvenida.
- Dirección de destino (vínculo): inicio.html.
- Lugar del contenido: main-frame.

#### Botón Iniciar sesión

- Texto estático: INICIAR SESIÓN.
- Texto dinámico: La sesión expira en una semana.
- Dirección de destino (vínculo): login.php.
- Lugar del contenido: main-frame.

#### Botón Administrar

- Texto estático: ADMINISTRAR.
- Texto dinámico: Sección restringida.
- Dirección de destino (vínculo): admin.html.
- Lugar del contenido: main-frame.

#### Botón Acerca de...

- Texto estático: ACERCA DE...
- Texto dinámico: Información de SICSS y contacto.
- Dirección de destino (vínculo): about.php.
- Lugar del contenido: main-frame.

El código para insertarlos es exactamente el mismo que se utilizó para insertar la animación de cabecera en el capítulo “4.2 ANIMACIÓN FLASH (CABECERA *TOP-FRAME*)”.

Una vez que se insertaron los botones con la configuración mencionada sólo hará falta posicionarlos del lado izquierdo de tal modo que utilicen uniformemente el espacio del marco izquierdo, el resultado de “Botones.html” se muestra a continuación:



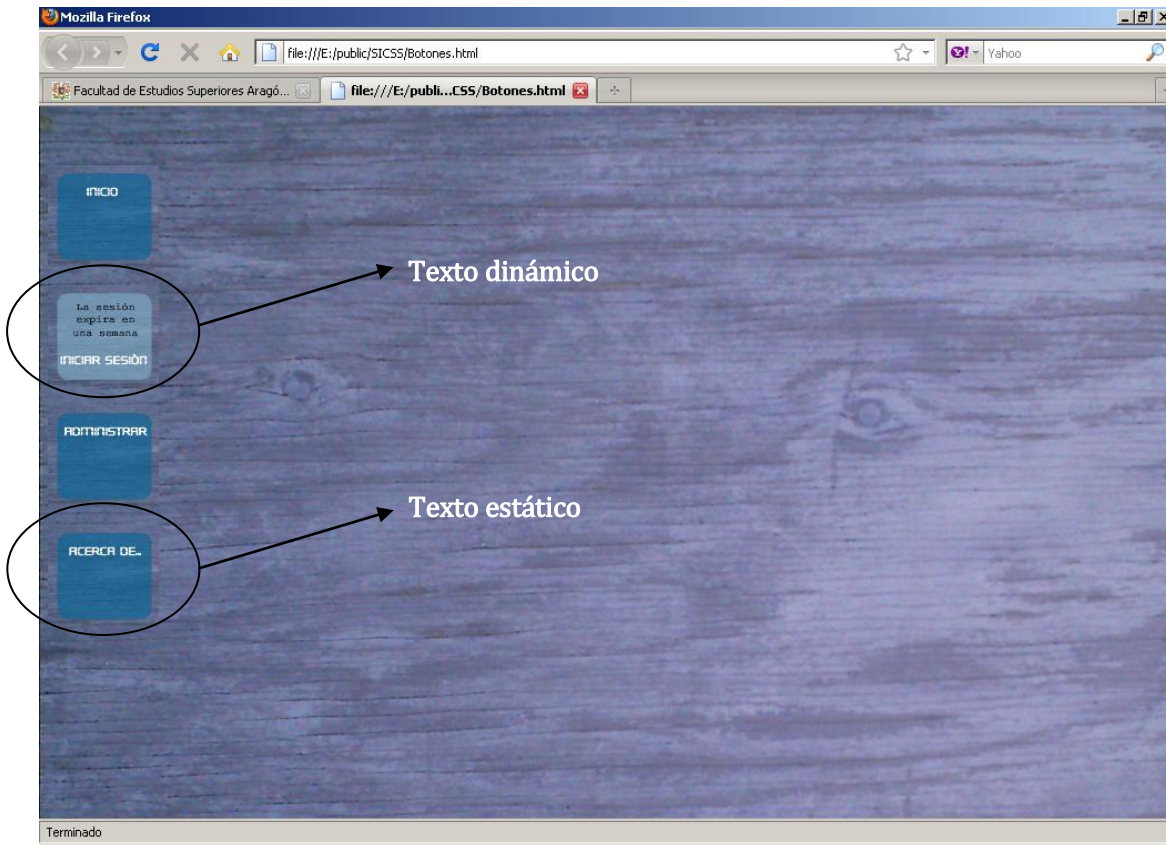


Figura 4.3 - Menú de botones animados de la página web.

Los botones contienen textos estáticos que son los que se muestran todo el tiempo mientras el cursor del mouse no los toque, y dinámicos que son aquellos que se podrán apreciar cada vez que el mouse se deslice encima de ellos. Dichos botones siempre estarán ahí a pesar que se estén realizando operaciones dentro del sistema, al igual que la animación, ya que todos los contenidos se mostrarán del lado derecho.

## 4.4 MARCO DE CONTENIDO (MAIN-FRAME)

### 4.4.1 SECCIÓN DE “INICIO”

A partir de aquí todas las demás páginas, módulos y resultados se hallarán en “main-frame” o marco principal. En este apartado se explicará el contenido de la página de inicio la cual sólo contiene una bienvenida al sistema, misma que aparecerá cada vez que se presione el botón “INICIO” del menú izquierdo.

Esta sección aún pertenece al diseño, y como sabemos este proyecto está enfocado a la parte operativa por lo que no daré detalle del código empleado para algunas decoraciones que se han agregado.

Inicialmente colocaré como imagen de fondo el logotipo de la UNAM, pondré un texto titular que dé la bienvenida al usuario: “Bienvenido al sistema de administración y conteo para prestadores de servicio social en la FES Aragón” y como centro la fecha de hoy, acompañada de un reloj que previamente he desarrollado con el lenguaje JavaScript cuyo código es muy extenso y no hay necesidad de explicarlo.

Finalmente un pequeño dialogo colocado en la parte de abajo que ofrece la instalación de “Adobe Flash Player®” debido a que existe la posibilidad que sea la primera vez que se abre una página de internet en esa computadora y por lo tanto no tenga instalado este plugin (pequeño complemento del navegador para reproducir ciertas partes de una página), por lo que no podría visualizarse en pantalla la animación de la cabecera y los botones Flash de la izquierda, al darle clic a dicho elemento la página direccionará automáticamente a la descarga e instalación.

Y el resultado de “incio.html” es el siguiente:

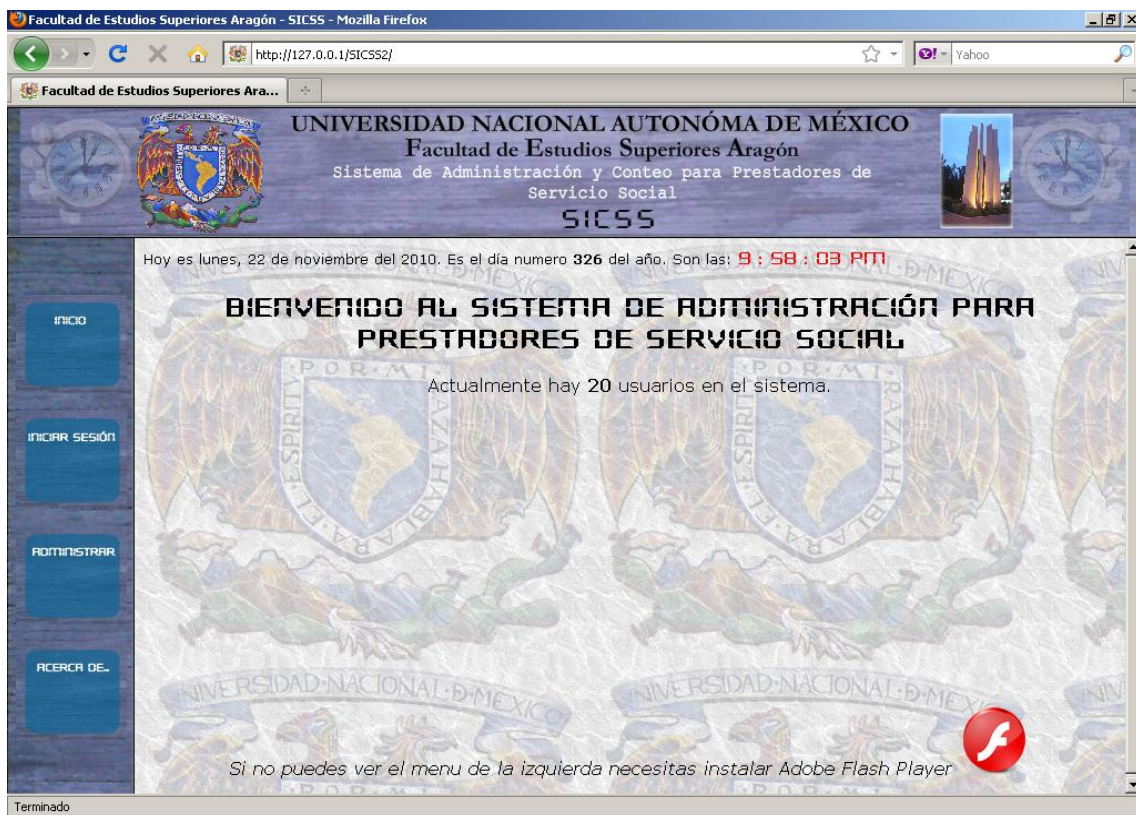


Figura 4.4 - Menú de botones animados de la página web.

Esta sección es sólo para dar bienvenida al usuario, informando la fecha y tiempo actuales y el número de usuarios en el sistema (asesores y prestadores de servicio), por

lo que las secciones más importantes y con funcionalidades se mencionan a continuación.

#### 4.4.2 SECCIÓN DE “ADMINISTRAR”

Para comenzar a utilizar el sistema es necesario contar mínimamente con un asesor registrado, este procedimiento será llevado a cabo por el administrador o responsable del área de informática.

En esta sección se podrán hacer varios ajustes a la página para controlar el contenido y el posterior uso de los usuarios, tanto del personal que requiere a los servicios sociales, como al prestador de servicio. Las operaciones posibles a realizar son las siguientes:

- Registrar asesores y usuarios.
- Consultar asesores y usuarios registrados, usuarios en línea e historial de uso de los mismos.
- Modificación de datos de asesores, usuarios y total de horas de un usuario.
- Agregar o quitar lista de lugares y computadoras (IP) que pueden ingresar al sistema.
- Configuración del “Sistema de Restricción de IP”.

Dichos rubros estarán acomodados en un menú que previamente he diseñado con ayuda de varias herramientas. Para entrar en esta sección se deben tener las contraseñas administrativas debido a que es una página restringida.

##### 4.4.2.1 Menú de administradores

Una vez que el administrador ha entrado al sistema se desplegará en la parte superior un menú: “menujs.html (menú JavaScript)”; éste se encargará de dibujar sub-menús al pasar el mouse por encima, el esquema en HTML es el siguiente

#### REGISTRAR

- . Asesor
- . Usuario

#### CONSULTAR

- . Historiales
- . Usuarios en línea
- . Usuarios registrados
- . Asesores

## MODIFICAR

- . Datos de asesor
- . Datos de usuario
- . Horas de usuario
- . IP
- . Lugar

## CONFIGURACIÓN

- . IP

Este menú estará desarrollado mediante la propiedad de “listas” en HTML, para éste caso se utilizará una lista desordenada `<ul>` ya que no es necesario darle orden a las opciones en cuanto a incisos se refiere. El código se muestra a continuación:

```

1. <ul id="nav">
2. <li><a id="checked" href="#">Registrar</a>
3. <ul>
4. <li><a href="formreg.php">Asesor</a></li>
5. <li><a href="formregs.php">Usuario</a></li>
6. </ul>
7. </li>
8. <li><a id="checked" href="#">Consultar</a>
9. <ul>
10. <li><a href="historiales.php">Historiales</a></li>
11. <li><a href="consultas.php">Usuarios en l&iacute;nea</a></li>
12. <li><a href="lista_us.php">Usuarios registrados</a></li>
13. <li><a href="asesores.php">Asesores</a></li>
14. </ul>
15. </li>
16. <li><a id="checked" href="#">Modificar</a>
17. <ul>
18. <li><a href="edit_ase.php">Datos de asesor</a></li>
19. <li><a href="edit_usu.php">Datos de usuario</a></li>
20. <li><a href="edit_hrs.php">Horas de usuario</a></li>
21. <li><a href="edit_ip.php">IP</a></li>
22. <li><a href="edit_lug.php">Lugar</a></li>
23. </ul>
24. </li>
25. <li><a id="checked" href="#">Configuraci&oacute;n</a>
26. <ul>
27. <li><a href="conf_ip.php">IP</a>
28. </ul>
29. </li>
30. </ul>

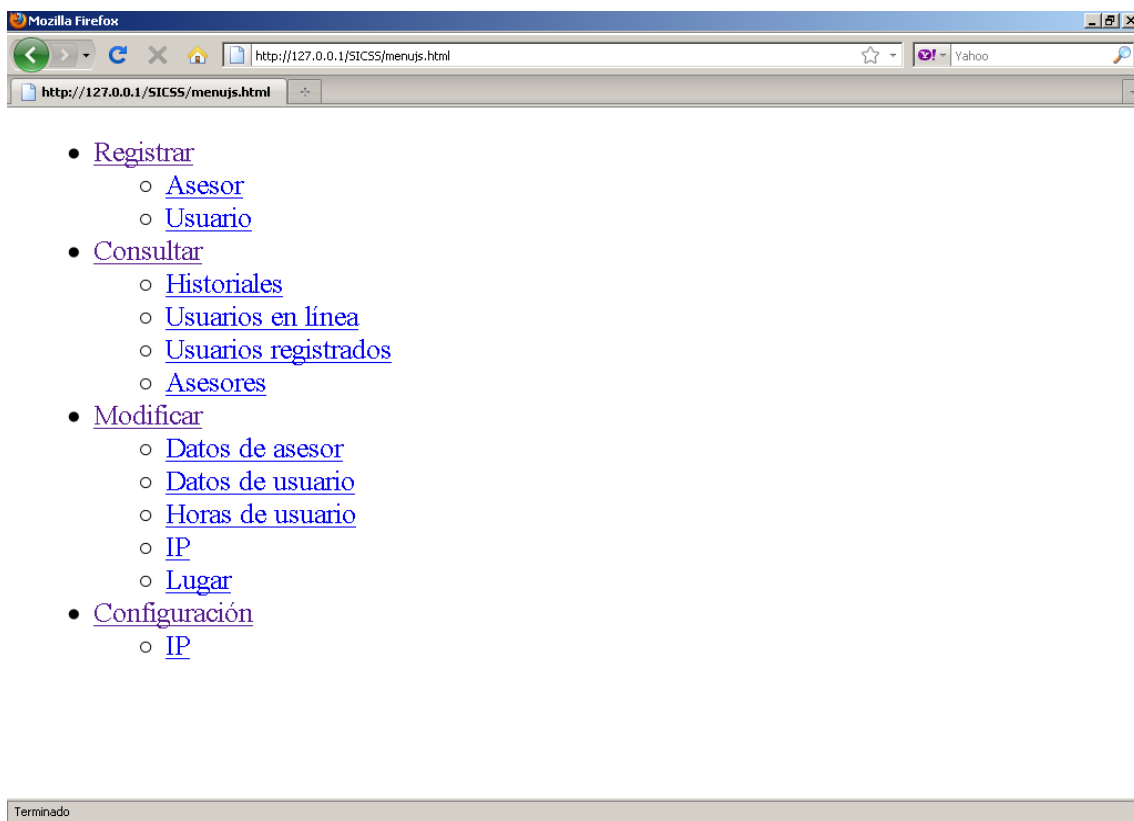
```

El orden de cada opción se mostrará de acuerdo a la posición en que se encuentran en el código, y cada una de las opciones mencionadas en este menú harán referencia a los diferentes archivos que también se nombran ahí.

Por ejemplo: “Consultar” es el título del menú, e “Historial” es la sub-opción oculta que aparecerá al pasar el mouse por encima. Al dar clic se direccionará al modulo “historial.php”.

Los encabezados de cada menú están dentro de una lista independiente y las sub-opciones en otra para así manipular los encabezados como títulos que se mostrarán y las sub-opciones como menús ocultos.

El resultado de este menú aun sin formato de diseño se muestra a continuación:



*Figura 4.5 - Menú de administradores diseñado en HTML.*

Cuando se quiere optimizar la apariencia de algún elemento HTML nos apoyamos en las hojas de estilo en cascada (CSS), ya anteriormente se ha agregado un archivo “estilos.css” vacío, sin embargo será utilizado en todos los módulos y páginas del SICSS, por lo que no será necesario hacer todo un apartado para él, sino que se irá actualizando su contenido a lo largo del capítulo.

Para este menú de usuarios se aplicará una técnica especial para apoyar al diseño del CSS, gracias a un framework que personalmente defino como “librería” ya que su función es incorporar más y/o mejores propiedades de las comunes a algún lenguaje de

programación (en este caso a JavaScript), llamada “jQuery”. Con ella lograré convertir el menú mostrado anteriormente a uno animado y atractivo a la vista.

Lo primero que hay que hacer es identificar a cada elemento, títulos y sub-opciones, para este caso ya coloqué un “id” (identificador) a los títulos (nav y titulo), y a las sub-opciones se les identificará por su etiqueta <li>. Después con las propiedades del lenguaje CSS, que automáticamente lee cualquier navegador (así como el HTML y JavaScript), indicamos qué elemento del proyecto queremos editar, y posteriormente los atributos que queremos asociar, por ejemplo para darle un color de fondo y un color al texto se escribiría el siguiente código:

```
1. #nav li #titulo
2. {
3.   background:url(color.gif) top left repeat-x;
4.   color:#ffffff;
5. }
```

En la primera línea hago referencia a los identificadores de los títulos para posteriormente ponerles una imagen de fondo, que previamente he preparado y almacenado en el archivo color.gif (línea 3), y la línea 4 indica el color del texto en hexadecimal, en este caso utilicé el equivalente a blanco. El código fuente del lado del servidor como PHP y consultas a la base de datos, no serán mostradas por razones de seguridad.

El menú ya incorporado con su respectiva sección, botones de acceso y cabecera se muestran a continuación:

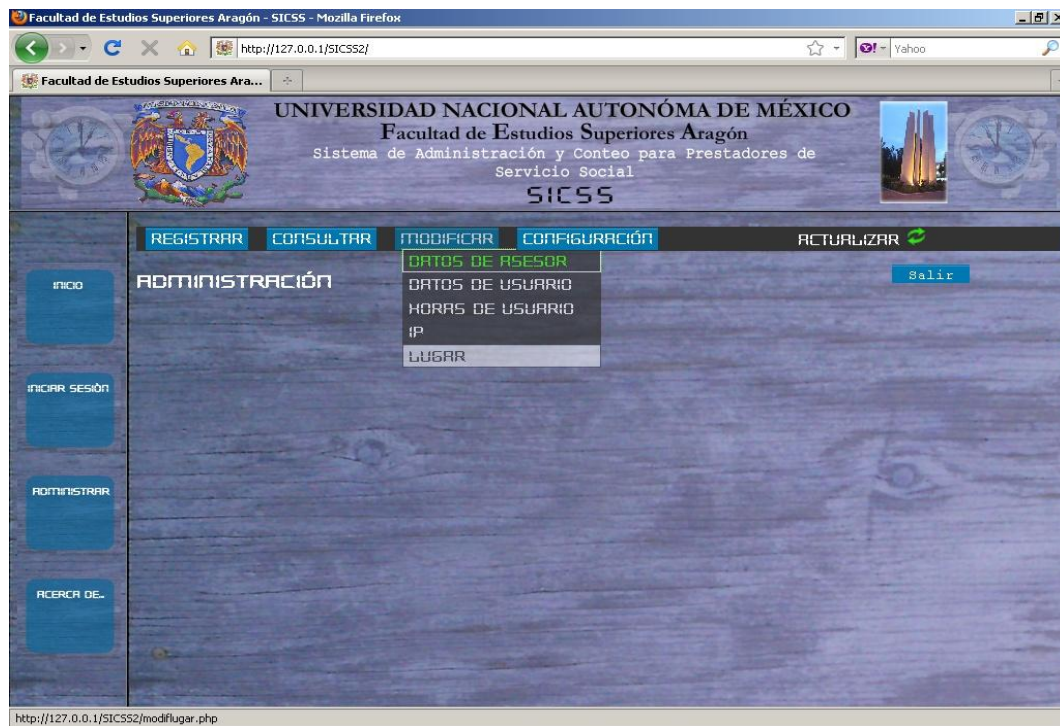


Figura 4.6 - Menú de administradores editado con CSS y JavaScript.

Cada vez que el mouse se deslice por los títulos azules se desplegará el menú en negro semitransparente, al mover el mouse sobre las nuevas opciones éstas contrastarán el color, y al tener una opción activa el texto tornará a color verde.

#### 4.4.2.1 Registros

Cuando se introduzca el sistema por primera vez es obligatorio registrar mínimamente a un asesor para comenzar a hacer uso del SICSS, de este modo se consigue adjuntar a una sesión de asesor a todos los prestadores de servicio social que se deseen dar de alta en el sistema y que pertenezcan al área del mismo.

El administrador del sistema es el único con privilegios para agregar asesores, por lo que cada vez que se requiera realizar esta operación se deberá acudir a él. La operación es simple y hasta cierto punto “clásica”, esto es porque se trata únicamente de llenar un formulario con algunos datos de la persona que se va a registrar.

Cabe resaltar que para llegar a este formulario el administrador ya debió entrar al sistema desde el apartado de “Administrar” y con las contraseñas administrativas proporcionadas al departamento de informática. Una vez ahí, se contará con un menú donde estarán disponibles las operaciones mencionadas anteriormente, para registrar a un asesor se debe ir al menú desplegable “Registrar → Asesor”, los detalles de la realización del menú se expondrán más adelante (en el menú de usuarios). Entonces, se hallará el siguiente formulario:

Facultad de Estudios Superiores Aragón - SICSS - Mozilla Firefox

http://127.0.0.1/SICSS2/

Facultad de Estudios Superiores Ara...

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
Facultad de Estudios Superiores Aragón  
Sistema de Administración y Conteo para Prestadores de Servicio Social  
SICSS

REGISTRAR CONSULTAR MODIFICAR CONFIGURACIÓN ACTUALIZAR

### REGISTRO DE ASESORES

**Datos para inicio de sesión**

Pseudónimo (*sin espacios)	<input type="text"/>
Contraseña	<input type="password"/>
Confirmar contraseña	<input type="password"/>

**Información personal**

Nombre(s)	<input type="text"/>
Apellido Paterno	<input type="text"/>
Apellido Materno	<input type="text"/>

**Seguridad**

Lugar	FUNAM sala 1
Dirección IP (opcional)	<input type="text"/>

Enviar

Terminado

Figura 4.7 – Formulario de registro para asesores.

Dónde todos los datos son obligatorios excepto la dirección IP, ya que ésta no necesariamente debe ser incluida en este momento.

Una vez que se ha dado clic al botón “enviar” aparecerá una pantalla de confirmación indicando si el asesor fue dado de alta exitosamente o si ocurrió algún error durante el proceso, esta última opción puede suceder si se dejaron campos obligatorios en blanco, si el pseudónimo ya está siendo usado por otro asesor, si la IP introducida ya ha sido registrada anteriormente, si la contraseña no coincide en ambos campos donde se solicitó, o sencillamente porque se está tratando de registrar a un asesor que ya ha sido registrado anteriormente.

Si lo que se requiere es registrar a un usuario, la tarea puede ser realizada por el administrador del sistema o bien por el asesor del área que ya ha sido registrado, con la única diferencia que el administrador puede registrar al usuario en cualquier departamento de la universidad, y el asesor registra únicamente en su departamento asociado.

Para llevarlo a cabo desde la sección de administración se cuenta con la opción dentro del menú desplegable “registrar → usuario” para visualizar el formulario con los campos donde se requiere la información del prestador de servicio social, para entrar desde la sección de asesores se debe ir al menú “asesor → registrar usuario”, de este modo es posible que el usuario se registre en el sistema por su propia cuenta.

El formulario se muestra a continuación:

The screenshot shows a web browser window with the URL `http://127.0.0.1/SICSS2/`. The page header includes the university name and the system title. A navigation bar contains buttons for 'REGISTRAR', 'CONSULTAR', 'MODIFICAR', 'CONFIGURACIÓN', and 'ACTUALIZAR'. The main content area is titled 'REGISTRO PARA PRESTADOR DE SERVICIO SOCIAL' and contains the following form fields:

Datos para utilización del sistema	
Nombre de usuario	<input type="text"/>
Contraseña	<input type="password"/>
Confirmar contraseña	<input type="password"/>

Información personal	
Número de cuenta	<input type="text"/>
Nombre(s)	<input type="text"/>
Apellido Paterno	<input type="text"/>
Apellido Materno	<input type="text"/>

Licenciatura:

Lugar:

**Enviar**

Figura 4.8 - Formulario de registro para prestadores de servicio social.



Cada vez que un usuario termine su registro se le proporcionaran algunos datos informativos como la fecha de término de servicio social estimada (tomando en cuenta que en promedio se realicen 4 horas de labor por día durante los días hábiles), las normas legales que establece el reglamento de servicio social de la UNAM, en cuanto a tiempo se refiere y la sugerencia de realizar una carta de extensión de servicio social en caso que exceda el tiempo estimado (6 meses).

Al igual que los registros de asesores, aquí se puede encontrar algún error en el registro, tales como: nombre de usuario repetido, contraseñas no coincidentes, usuario registrado con anterioridad o envío de campos vacíos.

#### 4.4.2.3 Consultas

Si en algún momento se requiere saber información de las personas que utilizan el sistema, aquí se hallarán las opciones para visualizar la información deseada, tal como:

- Consulta de asesores o usuarios registrados
- Consulta de usuarios en línea
- Consulta de historiales de uso

Cada vez que se elija alguna opción para consultas, como bien lo dice su nombre, únicamente se podrá hacer una consulta, es decir observar la información. Estas herramientas son útiles cuando se tiene a muchos usuarios registrados y se desea conocer la información detallada de alguno de ellos, o bien saber si algún usuario está registrado.

Al consultar a los **asesores** (opción disponible solo para administradores) se podrá saber si éstos se encuentran conectados al sistema o si no están haciendo uso de él, así como sus respectivos datos:



Figura 4.9 - Consulta de asesores registrados.

Al seleccionar esta opción se desplegará una tabla HTML, donde el servidor de bases de datos imprimirá todos los datos relacionados con el tipo de usuario en cuestión. Esta tarea la lleva a cabo un módulo específico de PHP que para el caso de usuarios registrados, también se invoca a “busqueda.php” si es que se necesita buscar a una persona en particular.

El algoritmo empleado para búsquedas no es complejo debido a la baja cantidad de registros con las que se va a trabajar, solo utilizaré una sentencia SQL para que el gestor de bases de datos seleccione a los registros según alguna variable introducida por el usuario que desea consultar. Por ejemplo, se tienen 15 registros de **prestadores de servicio social** y yo como administrador necesito saber si el alumno con el número de cuenta ‘123456789’ se encuentra registrado en el SICSS; el proceso sería entrar a la sección de administrar, elegir “Consultar → Usuarios registrados” del menú desplegable, y se mostrará la lista de usuarios registrados en donde tendremos la opción de buscar algún registro por medio del número de cuenta:



Figura 4.10 - Consulta de usuarios registrados.

Quizá para este ejemplo no aplicaría una búsqueda así ya que son muy pocas entradas y pueden leerse sin dificultad, sin embargo buscar a un solo usuario dentro de 250 registros o más puede resultar un tanto tedioso, por lo que se cuenta con la opción de búsqueda a través del número de cuenta, sólo se debe escribir en el campo de búsqueda ubicado en la parte superior derecha, para que el sistema muestre los resultados al instante.

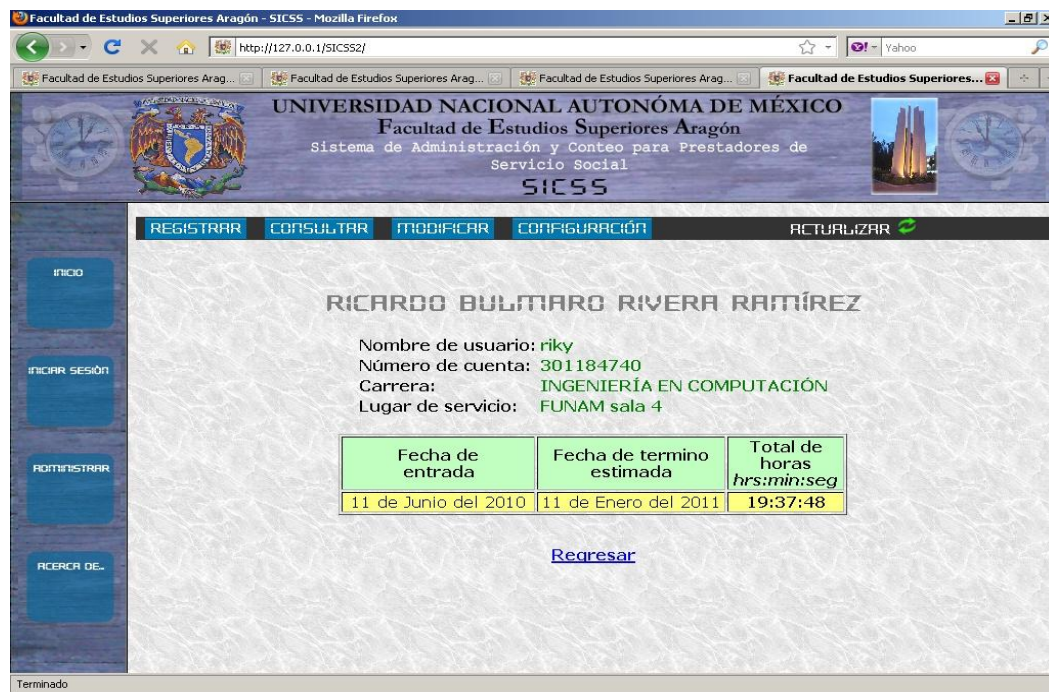


Figura 4.11 - Consulta individual a datos del usuario a través del número de cuenta.

Realizar consultas no es una tarea complicada, mucho menos para un administrador del sistema, no obstante este no es un proceso confidencial ya que se requiere que los asesores puedan consultar a sus respectivos servicios sociales, opción agregada en el menú desplegable “Consultar”, de la sección de asesores que aún está por verse, que funciona del mismo modo que el ya mencionado. También puede servir en caso que personas externas a la universidad quieran validar la estancia de algún estudiante como prestador de servicio social. Para ello se cuenta con la opción de iniciar sesión (en el apartado de “Iniciar Sesión”) como invitado donde únicamente tendrá acceso a la consulta de usuarios registrados y en línea.

La consulta a **usuarios en línea** lleva un proceso idéntico al de usuarios con la diferencia que el gestor de bases de datos PostgreSQL seleccionará únicamente a las personas que hayan iniciado su conteo de tiempo en ese momento, es decir a los de la tabla temporal “sesiones”; de no encontrarse ningún usuario activo, se enviará un mensaje con tal información. Para utilizar esta consulta se debe elegir el menú desplegable “Consultar → Usuarios en línea”.



Figura 4.12 - Consulta a la información de los usuarios que han iniciado su conteo.

A un costado derecho del menú principal, se cuenta con la opción “Actualizar”, a la cual con darle un clic de mouse refrescará la pantalla que esté mostrándose en ese momento; no es lo mismo darle “Actualizar” desde el navegador, de hacerlo estaríamos

actualizando el contenido de "index.php" lo cual llevaría nuevamente a la sección de inicio.

Finalmente tenemos las consultas de **historiales de uso**, que no son más que todas las entradas que tuvieron los usuarios cada vez que iniciaron su conteo de tiempo, al ser esta la tabla con mas registros, es evidente que se requiere de un buscador el cual funcionará de igual modo por medio de un "select" de SQL para consultar por números de cuenta y poder enlistar los registros de un solo usuario.

Estos datos están contenidos en la tabla "historiales" misma que se actualiza cada vez que el usuario inicia su conteo de tiempo. El formato en que se mostrarán los resultados se muestran a continuación.

Número de cuenta	Nombre completo	Carrera	Lugar de servicio	Fecha de entrada	Hora de entrada	Hora de salida	Duración de sesión hrs:min:seg
123456789	ALDO GÓMEZ ESCOBAR	ARQUITECTURA	FUNAM sala 1	2010-10-06	22:38:34	23:46:14	1:07:40
987654321	JORGE GARCÍA SALINAS	ARQUITECTURA	INFORMÁTICA	2010-10-06	22:38:53	23:46:08	1:07:15
301184740	RICARDO BULMARO RIVERA RAMÍREZ	INGENIERÍA EN COMPUTACIÓN	FUNAM sala 4	2010-10-06	22:31:14	23:46:00	1:14:46
301184740	RICARDO BULMARO RIVERA RAMÍREZ	INGENIERÍA EN COMPUTACIÓN	FUNAM sala 4	2010-09-27	22:23:35	23:26:48	1:03:13
301184740	RICARDO BULMARO RIVERA RAMÍREZ	INGENIERÍA EN COMPUTACIÓN	FUNAM sala 4	2010-09-05	20:56:09	21:28:35	0:32:26
301184740	RICARDO BULMARO RIVERA RAMÍREZ	INGENIERÍA EN COMPUTACIÓN	FUNAM sala 4	2010-07-18	14:37:19	15:09:42	0:32:23
	RICARDO						

Figura 4.13 - Consulta a registros con información de cada inicio de conteo de tiempo.

Para mostrar los registros de un usuario en específico se debe teclear el número de cuenta del mismo en la parte superior derecha en el campo de búsqueda y dar enter o clic al botón "buscar". Esta sección también está disponible para los usuarios.

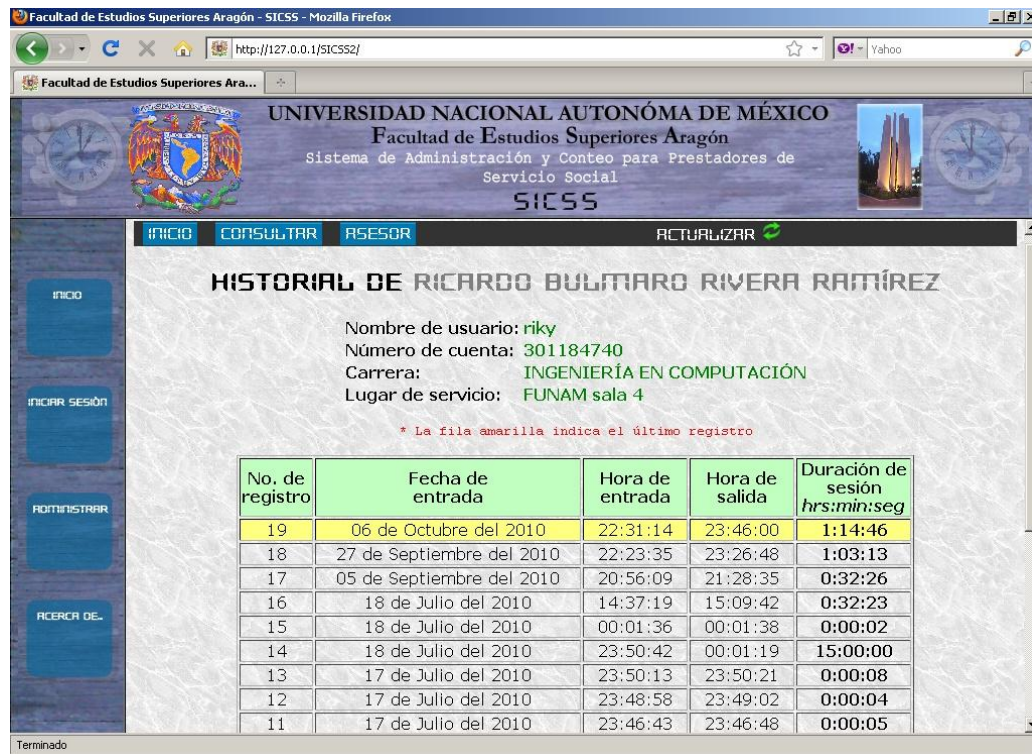


Figura 4.14 - Consulta individual de registros de cada inicio de conteo de tiempo.

#### 4.4.2.4 Modificación de datos

A veces es necesario realizar alguna modificación en los datos de alguna persona que se haya registrado, este proceso es exclusivo del administrador, por lo que si algún prestador de servicio social cometió un error en la inserción de sus datos, se deberán poner en contacto con el administrador del SICSS o directamente con el departamento de informática para realizar los cambios pertinentes.

Para utilizar esta herramienta se debe entrar como administrador desde el apartado “Administrar” y con las contraseñas administrativas, posteriormente elegir del menú desplegable “Modificar → Datos de usuario ó Datos de asesor”, para ambos casos se aplican los mismos criterios de los que ya se hablaron al momento de registrar por primera vez a un asesor o un usuario, la diferencia principal es que al entrar a este menú el sistema te desplegará una lista de personas registradas y se podrá elegir si quieres “Editar” los datos de tal registro o “Borrar” el registro tal como se muestra a continuación:



Figura 4.15 - Edición de registros de asesores.

Al editarlo aparecerá una pantalla donde los campos ya estarán llenos con la información que inicialmente se registró, por lo tanto sólo se deben hacer los cambios necesarios y dar clic en el botón “Guardar cambios”.

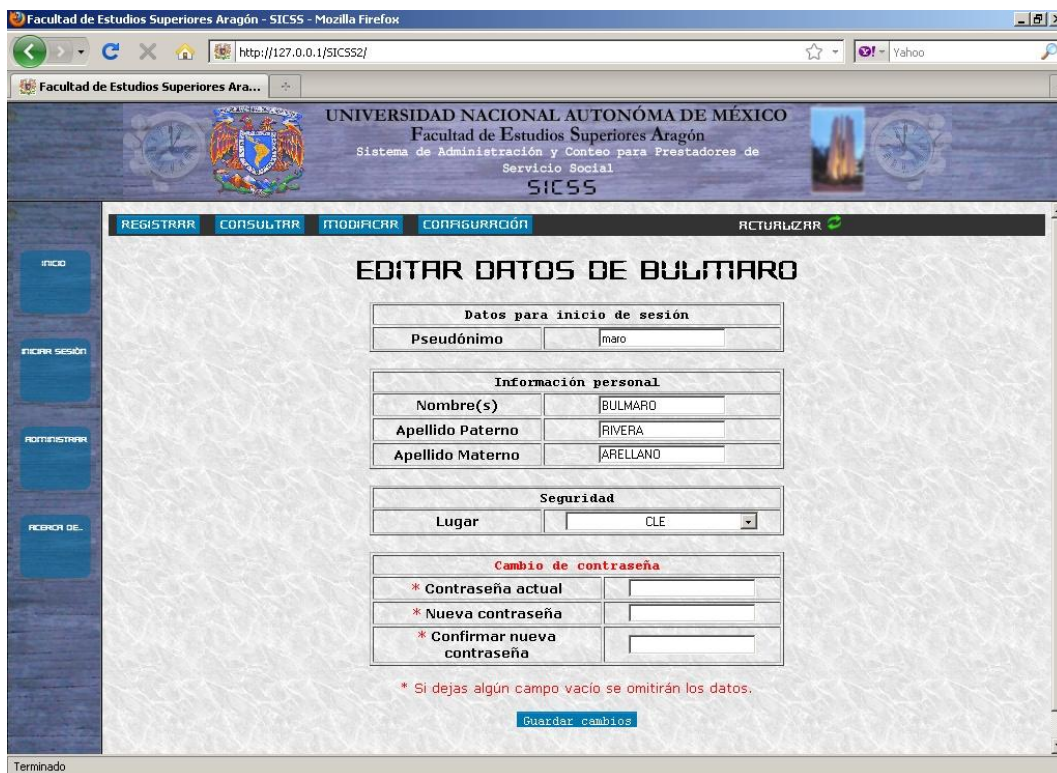


Figura 4.16 - Formulario para actualizar datos de asesores.

En caso que los prestadores de servicio requieran hacer un cambio de contraseña tendrán que llenar los campos que se aprecian en la parte inferior, sin embargo cada vez que se desee editar los datos de un usuario se verá esta opción por lo que se aclara que si se dejan en blanco los campos de contraseña no se aplicará ningún cambio a la misma.

Por otro lado, si se elige “Eliminar” a un registro, el sistema preguntará al administrador si está seguro de realizar esta acción ya que es irreversible, se debe ser cuidadoso con esta opción ya que si se elimina un registro se puede alterar el funcionamiento normal del SICSS, sin embargo se puede realizar esta acción para los usuarios que ya terminaron su servicio social, o bien para aquellos registros que fueron mal capturados.



Figura 4.17 - Borrado de un registro de asesor.

Pero hasta aquí se puede entender que esta sección es para cambiar datos que ya han sido establecidos de un supuesto modo correcto, pero también se deben contemplar algunos puntos importantes como: que algún usuario olvide terminar su conteo en el sistema, o que comience su labor del día sin iniciar su conteo, que se pierda la luz eléctrica, que el equipo de cómputo sufra una descompostura o simplemente que el asesor decida hacer una modificación en las horas laboradas del prestador de servicio social por razones que él conoce. Para solucionar esto se cuenta con la opción de “modificación de horas”, ubicar en el menú desplegable “Modificar → Total de horas” dentro de la sección “Administrar”, al entrar en este apartado podremos observar la siguiente pantalla:





Figura 4.18 - Consulta a registros de usuarios para editar su tiempo acumulado.

Como se puede apreciar aquí se despliega una lista con todos los registros existentes en la base de datos, se puede buscar manualmente al usuario y dar clic al botón “editar” correspondiente, o bien digitar su número de cuenta correspondiente en el buscador colocado en la parte superior derecha, de cualquier modo el sistema arrojará las horas totales junto con dos acciones a realizar, “Sumar” y “Restar”, bastará con ingresar la cantidad de tiempo a modificar con el formato: *h:mm* dónde: *h*= horas expresadas en 1 dígito (máximo 9), *mm*= minutos expresados en 2 dígitos (0 a 60). Una vez escritas las cantidades adecuadas solo se dará clic en el botón de la acción a realizar (sumar o restar) para alterar el valor en la base de datos.

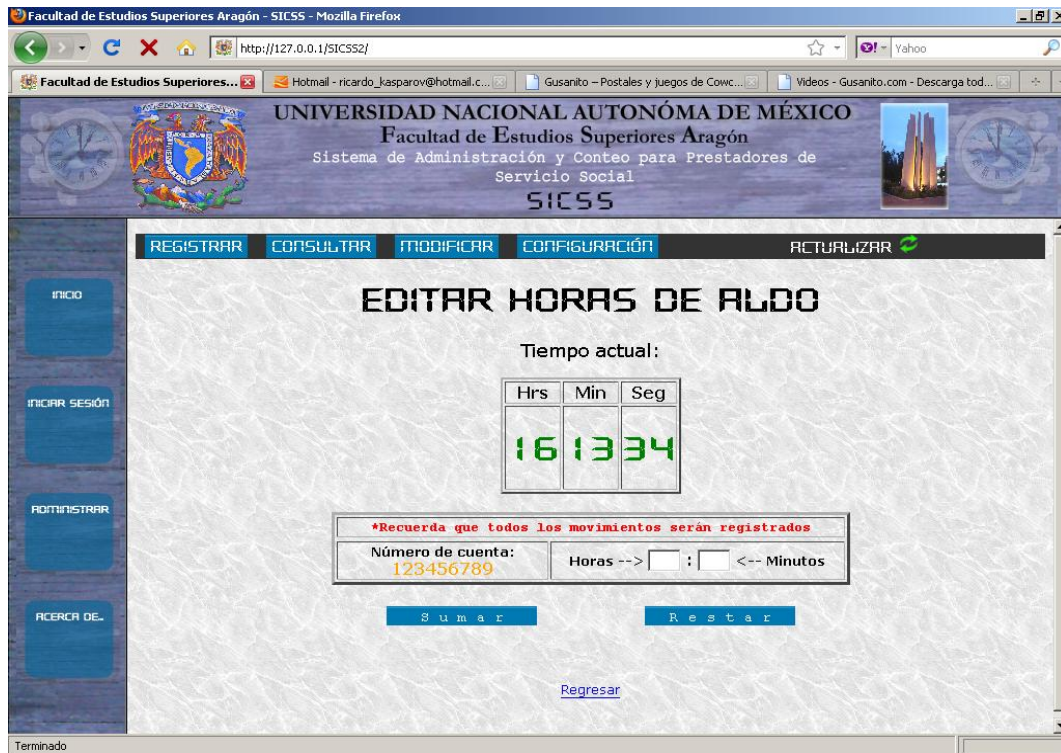


Figura 4.19 - Pantalla consecutiva a edición de horas de usuarios.

Al momento de sumar o restar alguna cantidad, el valor mostrado en “Tiempo actual” se actualizará automáticamente, y además de generar un registro en el historial del usuario (indicando que ese número de horas ha sido alterado por el administrador) también se dejará en ceros el valor de los segundos debido a su bajo nivel de importancia.

#### 4.4.2.5 Agregar o quitar lugares y direcciones IP

Este apartado es indispensable cuando se tiene habilitado el sistema de reconocimiento de IP (el cual será detallado más adelante), el procedimiento es sencillo pero de uso exclusivo del administrador de sistema. Para entrar a esta opción hay que moverse al menú desplegable “Modificar→IP”, al ingresar se despliega una lista con los lugares existentes y sus correspondientes direcciones IP asociadas:

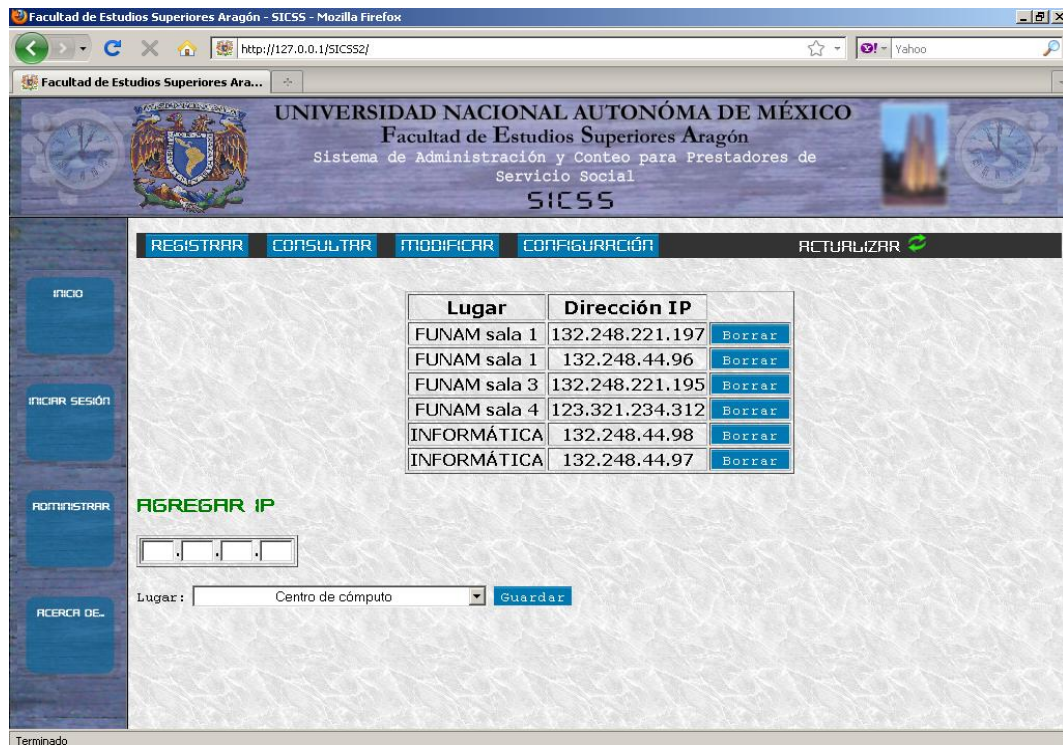


Figura 4.20 - Agregar o quitar direcciones IP y lugares de labor.

Si existe un lugar sin IP asignada no aparecerá en dicha lista; a un lado de cada registro estará disponible un botón con la etiqueta “Borrar”, que al presionarlo comprobará la autenticidad del clic preguntando al usuario si se está seguro de realizar tal operación, al dar clic en “Aceptar” se eliminará el registro permanentemente, por su contrario la opción “Cancelar” dejará intacta la pantalla.

Todas las direcciones que se enlisten en este apartado serán las únicas con el permiso de ingresar al sistema, por lo que cada vez que se agrega a un asesor se cuenta con la opción de agregar dos direcciones IP, en caso que se necesiten más se deberá hacer uso de esta sección.

En la parte final de la lista del lado izquierdo se encuentra la leyenda “Agregar IP” seguida de 4 campos correspondientes a los 4 segmentos de la IP a agregar, es aquí donde se debe teclear la IP a agregar así como el lugar en donde se encuentra la computadora con dicha dirección.

Si se agrega un IP que ya existe se mostrará un aviso indicando que dicha IP ya ha sido asignada así como el lugar al que pertenece, si se deja en blanco alguno de los campos sencillamente no se podrá realizar la operación.

#### 4.4.2.6 Configuración del Sistema de Restricción de IP

Esta sección es la más corta de todas, en cuanto a código y complejidad se refiere, ya que sólo dispone de una acción a realizar de la cual dependerá todo el SICSS, ésta se trata de una pantalla con la información del estado actual del sistema ya sea habilitado o deshabilitado. Si el sistema se encuentra habilitado entonces se verá un botón con la leyenda “Deshabilitar”, y viceversa.



Figura 4.21 - Sección para modificar estado de sistema de restricción de direcciones IP.

Cuando el sistema se encuentre deshabilitado no se tomará en cuenta ninguna dirección IP, esto significa que se podrá entrar al sistema desde cualquier computadora; dicha opción puede ser alterada únicamente por el administrador del sistema, pero cada vez que el sistema torne a “habilitado” entonces entrará en operación toda la base de datos existente con las direcciones IP registradas y aplicando las acciones mencionadas en el capítulo “4.4.2.4 Agregar o quitar lugares y direcciones IP”.

### 4.4.3 SECCIÓN DE “INICIAR SESIÓN”

En este apartado podrán iniciar sesión los asesores de cada área para que posteriormente sus prestadores de servicio social se registren o den de alta en el sistema e inicie el conteo de su tiempo, lo que hace a esta sección la más utilizada e importante del proyecto.

Al presionar el botón de “INICIAR SESIÓN” del menú izquierdo se direccionará a un archivo llamado “authcookie.php (autenticación de cookies)” donde se hará automáticamente una validación que comprueba si el asesor ya ha iniciado sesión anteriormente, para de este modo evitar que lo haga siempre que se presione este botón. Una vez iniciada la sesión el sistema abrirá la página de inicio restringida contenida en el archivo “loginas.php (logeado de asesor)”. De ser requerido se podrá cerrar la sesión en la página que muestra dicho archivo.

Al cerrar sesión se consigue destruir los cookies y las variables de sesión, esto significa que si se desea volver a ingresar se deberá escribir el nombre de usuario y contraseña del asesor nuevamente.

#### 4.4.3.1 Menú de usuarios

Una vez que el asesor ha iniciado sesión, “loginas.php” incluirá al archivo: “menujs.html (menú JavaScript)”; éste se encargará de dibujar en la parte superior izquierda del *main-frame* un menú con las nuevas opciones disponibles para el asesor y los prestadores de servicio social:

##### INICIO

- . Iniciar conteo
- . Terminar conteo

##### CONSULTAR

- . Historial
- . Usuarios en línea
- . Usuarios registrados

##### ASESOR

- . Registrar usuario
- . Forzar baja

A un lado del menú habrá un texto breve de bienvenida al asesor y un botón de “cerrar sesión” del cual ya se dio explicación, sobre el centro de esta misma página colocaré un reloj de tipo analógico que avanzará en tiempo real, y el logotipo de la UNAM.

El código, esquema y diseño de implementación de este menú es idéntico al mostrado en el capítulo “4.4.2.1 Menú de administradores”, sin embargo la única diferencia son las opciones expuestas anteriormente.

El resultado de la página de inicio con el fondo, logotipo de la UNAM, reloj analógico y un contador de visitantes se muestra a continuación:

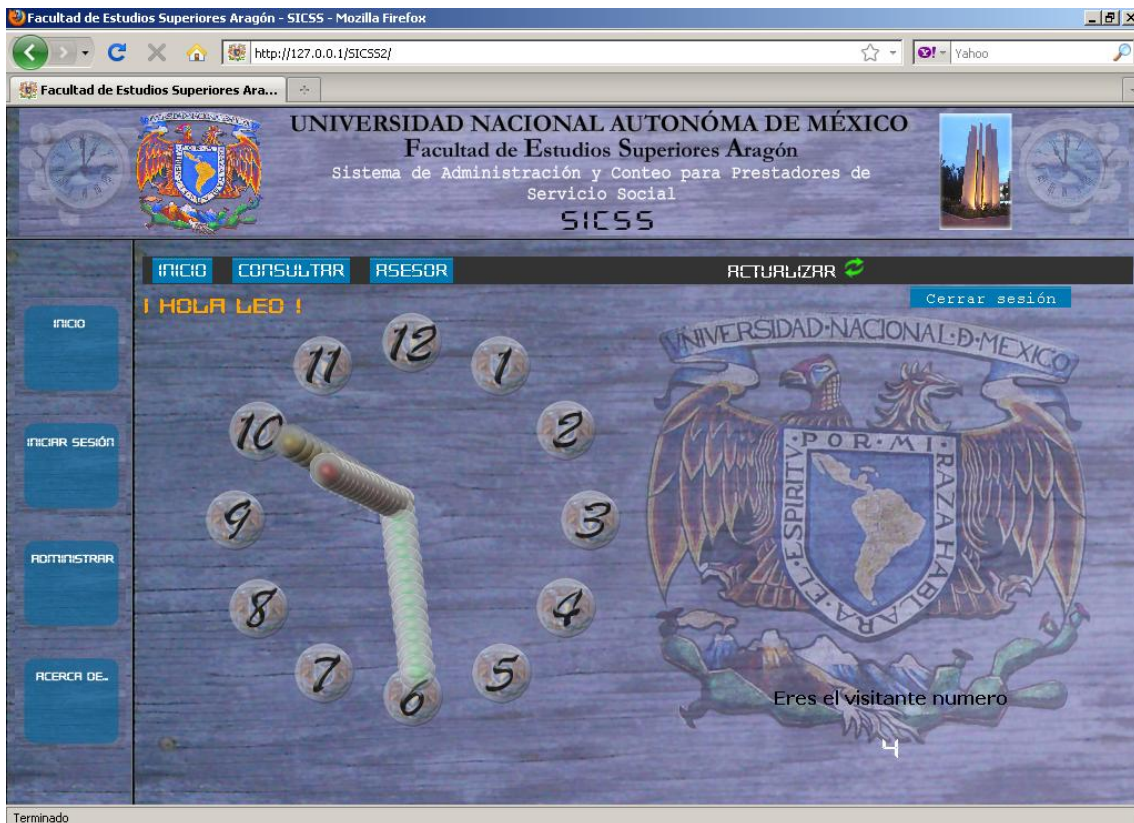


Figura 4.22 - Pantalla de bienvenida al inicio de sesión del asesor.

El contador de visitantes almacena direcciones IP en la base de datos y posteriormente las cuenta para así mostrar un número en pantalla. El algoritmo empleado tiene la peculiaridad de no contar al mismo usuario sino hasta que uno diferente entre al sistema.

Ahora me enfocaré en la parte operativa del SICSS de tal modo que explicaré cada una de las opciones del menú de usuarios citado anteriormente, pero en orden en que se deberían ir utilizando y no en el que están dibujados en pantalla.

### 4.4.3.2 Registrar usuarios

Cuando ya esté disponible esta página web para su uso en la Facultad, ya se deben haber realizado ciertas tareas por parte del administrador para que no tengan problemas con su uso, la más importante sin duda es registrar a las personas que vayan a solicitar prestadores de servicio social, es decir, los asesores. Este procedimiento ya fue explicado en el capítulo “4.4.3 Sección de Administrar” a diferencia que aquí la tarea la puede llevar a cabo el asesor y no necesariamente el administrador.

Para agregar al sistema a un nuevo prestador de servicio social se deberá dirigir al menú desplegable “Asesor → Registrar usuario”, del nuevo menú que aparece, con esto estaremos abriendo el archivo “form\_registro.php” que contiene un formulario que será mostrado en pantalla pidiendo los datos necesarios para registrar a un prestador de servicio social, tales como:

- . Número de cuenta
- . Nombre completo
- . Licenciatura
- . Lugar
- . Nombre de usuario
- . Contraseña

Para implementar un formulario que sea capaz de enviar los datos al servidor y sean almacenados se debe llevar una serie de pasos, lo inicial es dibujar en pantalla los campos correspondientes, para esto utilizaré las etiquetas <form></form> y dentro de ellas la etiqueta <input> con el tipo de campos que se desean imprimir tales como:

- Texto (text)
- Contraseña (password)
- Variables ocultas (hidden)
- Botones (submit)

Por ejemplo, para imprimir campos que pidan: nombre de usuario, contraseña y un botón de envío se emplearía el siguiente código:

1. <form method= "post" action="validar.php">
2. Nombre de usuario: <input type="text" name="nombre" /><br />
3. Contraseña: <input type="password" name="contra" /><br />
4. <input type="submit" name="boton" value="Enviar" />
5. </form>

Inicialmente se escribe la etiqueta <form> para definir el tipo de envío, el cual puede ser GET o POST, la diferencia entre ambas es que GET muestra el contenido de los datos enviados por el formulario en la URL, y POST los oculta; también se indica el archivo al

cuál se debe direccionar una vez que se ha dado clic en el botón “Enviar”, en este caso sería “validar.php”. En las líneas 2 y 3 se imprimen los campos con un nombre (name) para poder rescatar los datos desde el archivo “validar.php”, en la línea 4 se coloca el botón de envío el cual contiene el parámetro “value (valor)” que representa la etiqueta que tendrá dicho botón, finalmente se tiene el cierre de la estructura del formulario </form>.

Este tipo de código se empleará constantemente a lo largo del proyecto para pedir los datos necesarios al usuario cada vez que se requiera, el formulario de registro de usuarios se muestra a continuación:

The screenshot shows a web browser window with the URL `http://127.0.0.1/SICSS2/`. The page header identifies the institution as the Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Aragón, Sistema de Administración y Conteo para Prestadores de Servicio Social (SICSS). The main content area is titled 'REGISTRO PARA PRESTADOR DE SERVICIO SOCIAL' and contains the following form elements:

- Datos para utilización del sistema:**
  - Nombre de usuario:
  - Contraseña:
  - Confirmar contraseña:
- Información personal:**
  - Número de cuenta:
  - Nombre(s):
  - Apellido Paterno:
  - Apellido Materno:
- Licenciatura:
- Lugar:

At the bottom of the form is a blue button labeled 'Enviar'. A sidebar on the left contains navigation links: INICIO, INICIAR SESIÓN, ADMINISTRAR, and ACERCA DE... The status bar at the bottom left indicates 'Terminado'.

Figura 4.23 - Formulario de registro de usuarios.

Cuando los datos se hayan introducido correctamente deberá presionarse el botón con la leyenda “Enviar” para que el sistema valide los datos. Inicialmente se pasa por una función de JavaScript que se encarga de comprobar que ninguno de los campos se haya dejado en blanco y que las contraseñas sean idénticas, si la función encuentra algo diferente se enviará un mensaje al usuario informando del error y no se podrá continuar hasta que éste sea corregido.

He nombrado a la función como “validaAlta” y se encuentra codificada en un archivo que llame “fx.js (funciones)” y el código se muestra a continuación:



```

1. function validaAlta (a)
{
  a.nombre.value = a.nombre.value.toUpperCase();
  a.ap_pat.value = a.ap_pat.value.toUpperCase();
  a.ap_mat.value = a.ap_mat.value.toUpperCase();
  if(a.nick.value=="")
  {
    bAlert("Olvidaste escribir tu nombre de usuario");
    return false;
  }
  if(a.pass.value=="")
  {
    bAlert("Olvidaste escribir tu contrase&ntilde;a");
    return false;
  }
  if(a.vpass.value=="")
  {
    bAlert("Confirma tu contrase&ntilde;a para continuar");
    return false;
  }
  if(a.cuenta.value=="")
  {
    bAlert("Olvidaste escribir tu n&uacute;mero de cuenta");
    return false;
  }
  if(a.nombre.value=="")
  {
    bAlert("Olvidaste escribir tu nombre");
    return false;
  }
  if(a.ap_mat.value=="")
  {
    bAlert("Olvidaste escribir tu apellido materno");
    return false;
  }
  if(a.ap_pat.value=="")
  {
    bAlert("Olvidaste escribir tu apellido paterno");
    return false;
  }
  else
  if (a.pass.value != a.vpass.value)
  {
    bAlert ("Verifica que la contrase&ntilde;a coincida en ambos campos");
    return false;
  }
  else
  {
    return true;
  }
}

```

Como se puede apreciar el código es un tanto repetitivo, sin embargo es necesario hacerlo de este modo para conseguir mostrar mensajes que contengan el detalle exacto del error. Para posteriores usos de este tipo de validaciones se tendrá como base el código anterior para que no haya necesidad de repetirlo.

Es por esto que dicho archivo debe incluirse en las cabeceras de los módulos donde sea necesario su uso así como los estilos y frameworks, del siguiente modo:

1. `<style media="all" type="text/css"> @import "estilos.css";</style>`
2. `<style media="all" type="text/css"> @import "jquery_impromptu.css";</style>`
3. `<script lenguaje="javascript" src="jquery.txt"></script>`
4. `<script lenguaje="javascript" src="jquery_impromptu.js"></script>`
5. `<script language="JavaScript" src="fx.js"></script>`

En la línea 1 vemos la inclusión del ya nombrado archivo estilos.css, las líneas 2, 3 y 4 incluyen al framework jQuery y una librería adicional llamada jQuery Impromptu que sirve para realizar mensajes de alerta personalizados que serán utilizados para mostrar los respectivos mensajes que arroje la función validaAlta (contenida en la línea 5 en "fx.php") para el usuario, por ejemplo si se dejara vacío el campo que pide el nombre y se llenaran los demás, la función "validaAlta" mandaría a llamar a "jQuery.txt" que al mismo tiempo incluiría a "jQuery Impromptu" en su versión "js" y "css" para imprimir en pantalla una alerta personalizada de información al usuario. El ejemplo se muestra a continuación:

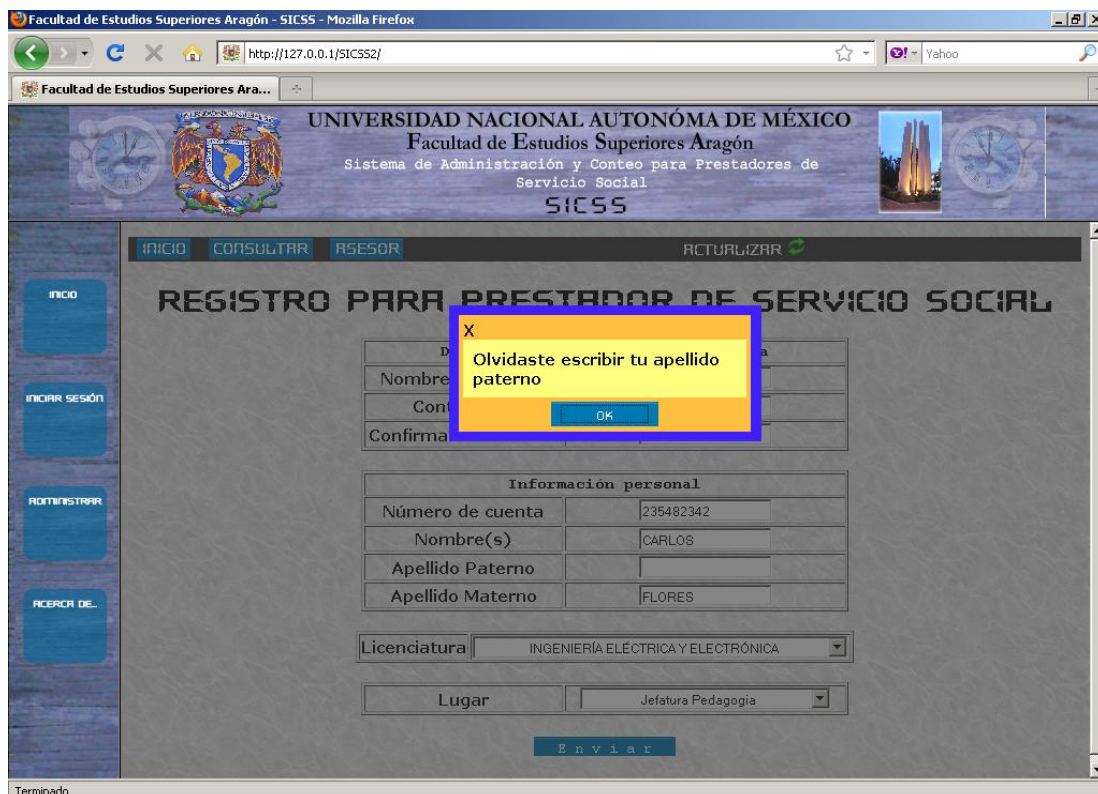


Figura 4.24 - Mensajes de alerta al dejar un campo de formulario en blanco.

En la línea 1 se declara el nombre de la función, de la 3 a la 6 se convierten los caracteres correspondientes al nombre completo en mayúsculas, para evitar que el usuario escriba forzosamente la primera letra en mayúscula de su(s) nombre(es) y apellidos; las siguientes líneas declaran un mensaje de alerta perteneciente a jQuery Impromptu para mostrarse cada vez que se encuentre un campo vacío, de ser así se colocará en pantalla el mensaje de alerta y no se permitirá avanzar, si no se encuentra ningún espacio en blanco se continuará con la validación ya en lenguaje servidor, es decir con PHP.

A pesar de la existencia de esta función, cabe la posibilidad que el navegador tenga deshabilitada la ejecución de procesos JavaScript, es por esto que se ha tenido que repetir la rutina de validación de espacios en blanco en PHP, encabezando a todo el proceso de validación para evitar realizar todas las comprobaciones que más adelante se citarán en caso de existir espacios en blanco.

Es evidente que no se debe dejar ningún campo en blanco ya que todos los datos deben ser capturados para el correcto funcionamiento del sistema, pero existen excepciones como es el caso de la sección de “Registro de asesores” que permite la omisión de escritura en el campo de “agregar dirección IP” ya que no es completamente necesario que ésta sea agregada en ese momento. Otro caso es la sección de mensajes, de la cual se hablará en la última parte de este capítulo (Sección “Acerca de”), dónde se puede hacer caso omiso de la introducción de un correo electrónico.

Quizá nace la pregunta: ¿Por qué entonces no sólo crear la función de validación de blancos en PHP? Y la respuesta está a la vista, literalmente, ya que JavaScript introduce mensajes difuminados, semitransparentes y con un alto nivel de animación mucho más innovador a la vista de lo que se puede hacer con PHP, éste sólo es capaz de mostrar los resultados del error en una página nueva y sin atributos de diseño atractivos, aunque se cuenta con un nivel de seguridad más alto por el simple hecho de ejecutarse en el servidor.

La función de validación en PHP es más simple y se muestra a continuación:

```

1. $cuenta = $_POST['cuenta'];
2. $nombre = $_POST['nombre'];
3. $ap_pat = $_POST['ap_pat'];
4. $ap_mat = $_POST['ap_mat'];
5. $carrera = $_POST['carrera'];
6. $lugar = $_POST['lugar'];
7. $nick = $_POST['nick'];
8. $pass= $_POST['pass'];
9. $vpass = $_POST['vpass'];
10. if($cuenta==""||$nombre==""||$ap_pat==""||$ap_mat==""||$carrera==""||$lugar==""||$nick==""
    ||$pass==""||$vpass=="")
11. {
12. echo "<h1>No se puede continuar</h1>";
13. echo "Dejaste algunos campos en blanco<br>";

```

- ```

14.     echo 'Presiona el botón "Atrás" de tu navegador, o da click <b><a
        href="javascript:history.back()">Aquí</a></b> para regresar.';
15. }

```

Primero se obtienen los datos del formulario contenidos en la propiedad “name” a través de la propiedad \$\_POST y se almacenan en variables, éstas están marcadas con el símbolo “\$” en las primeras 9 líneas. Inmediatamente después se define una condición para detectar a cualquiera de las variables que tenga por valor una cadena vacía, si se cumple, se mostrarán los mensajes escritos en las líneas 12 a 14 del código, y se detendrá la rutina de validación.

Esta última función estará contenida en un archivo llamado “regss.php (registro de servicio social)” que será invocado al momento de dar clic al botón “Enviar” del formulario de registro de usuarios. Ahora que sabemos el código que encabezará a regss.php, pero antes de pasar a desarrollar el resto de la autenticación se comienza la interacción con el sistema externo encargado de gestionar todos los datos que se almacenarán durante todos los módulos que lo requieran, me refiero a PostgreSQL, por lo tanto primero veremos tal configuración.

#### 4.4.3.2.1 Esquema de PostgreSQL

Antes de continuar con toda la sección de módulos de PHP, voy a crear el esquema que necesita utilizar la base de datos PostgreSQL. En la fase de análisis se captan todas las variantes que el sistema pueda tener a través de los casos de uso, es así como es más simple desarrollar estas secciones ya que todo está analizado, solo se necesita llevar a la práctica.

Lo primero después de instalar la base de datos “PostgreSQL 4.2.3” es crear un nombre de usuario y una contraseña, después se procede a crear una nueva base de datos. Estos datos serán referidos desde el archivo conexionBD.php que crearemos una vez terminado el esquema.

Ahora se agregan a la base de datos creada las tablas siguientes:

1. Asesores
2. Usuarios
3. Lugares
4. IPs
5. Passwords
6. Sesiones
7. Sesiones\_A
8. Historiales
9. ConfigIP
10. Visitantes

## 11. Mensajes

Toda tabla debe contener columnas o atributos, que son como los encabezados de un conjunto de información del mismo tipo, en el capítulo 2.2.1 se explican con claridad estos conceptos.

Se requiere configurar a cada una de las columnas dependiendo del tipo de datos que éstas vayan a almacenar, para ese proyecto se utilizarán cuatro:

- . Character varying.- Corresponde a datos con símbolos variables, este tipo será el más utilizado debido a su practicidad.
- . Date.- Tipo de dato exclusivo para almacenar fechas..
- . Integer.- Almacena números enteros (no decimales).
- . Serial.- Es un “Integer” especial que tiene la propiedad de autoincrementarse en 1 cada vez que se agrega un registro nuevo.

También se puede elegir si se quiere que la columna sea “Not null” o no, esto significa que la columna no puede quedar vacía, si es así el gestor de base de datos enviará un error en pantalla y no se almacenará correctamente la información. Y como último parámetro de configuración, se puede agregar un límite de tamaño al tipo de dato “character varying”, esta propiedad se aplica cuando se sabe el tamaño exacto del texto a introducir por el usuario para ahorrar memoria de almacenamiento.

### Tabla 1: “Asesores”

Esta tabla servirá para almacenar la información de los asesores, tendrá 6 columnas y todas con el tipo de dato “Character varying”, la propiedad “Not null”. A algunas columnas se les dará una limitación de tamaño para ahorrar espacio en el disco duro.

Las columnas son las siguientes:

- . Pseu: Pseudónimo o nombre de usuario que el asesor elija, este campo no puede repetirse con dos o más asesores ya que debe ser único debido a que será utilizado como valor de autenticación.
- . Pass: Contraseña del asesor, misma que será requerida al momento de iniciar sesión.
- . Lugar: Nombre del lugar en donde se hará uso del SICSS, por ejemplo el departamento de informática, alguna jefatura de carrera, el Centro de Lenguas Extranjeras (CLE), etc. Este dato servirá para asociarlo con el prestador de servicio social que labore en esa zona, para de este modo conseguir una mayor seguridad del sistema.
- . Nombre: Nombre o nombres del asesor.
- . Ap\_pat: Apellido paterno del asesor.
- . Ap\_mat: Apellido materno del asesor.

La configuración de las columnas se muestra a continuación:

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Pseu              | character varying | 25 caracteres         | No                         |
| Pass              | character varying | 25 caracteres         | No                         |
| Lugar             | character varying | Sin límite            | No                         |
| Nombre            | character varying | Sin límite            | No                         |
| Ap_pat            | character varying | Sin límite            | No                         |
| Ap_mat            | character varying | Sin límite            | No                         |

### Tabla 2: “Usuarios”

Esta tabla será útil para almacenar la información de los prestadores de servicio social, que para el sistema solo son usuarios. Las columnas se muestran a continuación:

- . Cuenta: Número de cuenta del prestador de servicio social, mismo que forzosamente debe ser único.
- . Nombre: Nombre o nombres del prestador de servicio social.
- . Ap\_pat: Apellido paterno del prestador de servicio social.
- . Ap\_mat: Apellido materno del prestador de servicio social.
- . Carrera: Licenciatura en curso del prestador de servicio social.
- . Lugar: Nombre del lugar o departamento en donde se laborará.
- . Inicio: Fecha de inicio del servicio social (generada automáticamente al momento del registro).
- . Terminó: Fecha de término del servicio social (generada automáticamente al momento del registro, ésta corresponde a 6 meses sumados a la fecha de inicio del servicio social)
- . Nick: Nombre de usuario que utilizará el prestador de servicio social cada vez que quiera iniciar su conteo de tiempo.

La configuración de las columnas se muestra a continuación:

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Cuenta            | character varying | 9 caracteres          | No                         |
| Nombre            | character varying | 30 caracteres         | No                         |
| Ap_pat            | character varying | 20 caracteres         | No                         |
| Ap_mat            | character varying | 20 caracteres         | No                         |
| Carrera           | character varying | Sin límite            | No                         |
| Lugar             | character varying | Sin límite            | No                         |
| Inicio            | Date              | No necesario          | No                         |
| Termino           | Date              | No necesario          | No                         |
| Nick              | character varying | 20 caracteres         | No                         |

**Tabla 3: “Lugares”**

Al momento de registrar asesores, el administrador debe tener el dato de en qué lugar se utilizará el sistema debido a que debe agregarlo, en la sección de “administrar”, antes de registrar al asesor. Para llevar a cabo el proceso es necesaria esta tabla.

Las columnas y configuración de la tabla “Lugares” se muestra a continuación:

- . Id: Identificador para distinguir a cada lugar con un valor numérico y ascendente.
- . Lugar: Nombre del lugar o departamento agregado.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Id                | Serial            | No necesario          | No                         |
| Lugar             | character varying | Sin límite            | No                         |

**Tabla 4: “IP”**

Esta tabla se implementará para registrar las direcciones IP de cada computadora en donde se desee personalizar el acceso único, es decir, que sólo se pueda usar el SICSS desde esa máquina para los usuarios asignados a ese lugar.

Las columnas y configuración de la tabla “IP” se muestra a continuación:

- . Id: Identificador para distinguir a cada IP con un valor numérico y ascendente.
- . Ip: Dirección IP.
- . Lugar: Nombre del lugar o departamento en donde está la computadora con dicha IP.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Id                | Serial            | No necesario          | No                         |
| Ip                | Character varying | 20 caracteres         | No                         |
| Lugar             | Character varying | Sin límite            | No                         |

**Tabla 5: “Passwords”**

Esta tabla tiene por objetivo almacenar los datos que tengan que ver con seguridad, principalmente la contraseña y horas acumuladas de los usuarios, a continuación se muestran las columnas y su respectiva configuración:

- . Nick: Nombre de usuario único elegido por el prestador de servicio social, esta columna servirá para relacionarla con la tabla “Usuarios”.
- . Pass: Contraseña del usuario.
- . Horas: Total de horas acumuladas del usuario.
- . Cuenta: Número de cuenta del usuario, esta columna será útil para relacionarla con las tablas “Sesiones” e “Historial”, mismas que serán explicadas mas adelante.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Nick              | Character varying | 20 caracteres         | No                         |
| Pass              | Character varying | 20 caracteres         | No                         |
| Horas             | Character varying | 9 caracteres          | No                         |
| Cuenta            | Character varying | 9 caracteres          | No                         |



**Tabla 6: “Sesiones”**

Una vez que un prestador de servicio social ha iniciado su conteo de tiempo el sistema debe almacenar temporalmente un registro con la información necesaria para saber si se tiene activo su inicio de conteo de tiempo en el sistema y datos de entrada como fecha y hora para generar un historial.

Las columnas y configuración de esta tabla se muestran a continuación:

- . Fecha: Fecha de entrada del usuario.
- . Cuenta: Número de cuenta del usuario, que servirá como identificador en la tabla.
- . Hora: Hora de entrada del usuario.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Fecha             | Date              | No necesario          | No                         |
| Cuenta            | Character varying | 9 caracteres          | No                         |
| Hora              | Character varying | 9 caracteres          | No                         |

**Tabla 7: “Sesiones\_A”**

También se va a requerir una tabla para almacenar ahora la sesión que halla iniciado el asesor correspondiente, esto con la finalidad de controlar el acceso en posteriores entradas al sistema, asociar cada registro del prestador de servicio social a su respectivo asesor y lugar de labor, y conocer el número total de usuarios (tanto prestadores como asesores) dentro del sistema.

Las columnas y configuración de esta tabla se muestran a continuación:

- . Fecha: Fecha de entrada del asesor.
- . Pseudo: Pseudónimo que servirá como identificador del asesor en la tabla.
- . Hora: Hora de entrada del asesor.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Fecha             | Date              | No necesario          | No                         |
| Cuenta            | Character varying | 9 caracteres          | No                         |
| Hora              | Character varying | 9 caracteres          | No                         |

**Tabla 8: “Historiales”**

Esta tabla se auto rellena cada vez que un usuario termine su conteo de tiempo en el sistema, estos datos serán permanentes y servirán para consultar todas las entradas de cada prestador de servicio social.

Las columnas y su respectiva configuración se muestran a continuación:

- . Id: Contador automático de registros, utilizado para ordenar las entradas conforme se van almacenando.
- . Cuenta: Número de cuenta del usuario.
- . Fecha\_entrada: Fecha en que se inició el conteo de tiempo en el sistema.
- . Hora\_entrada: Hora en que se inició el conteo de tiempo en el sistema.
- . Hora\_salida: Hora en que finalizó el conteo de tiempo en el sistema.
- . Horas\_sesion: Tiempo exacto que el usuario utilizó el sistema en una sesión.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Id                | Serial            | No necesario          | No                         |
| Cuenta            | Character varying | 9 caracteres          | No                         |
| Fecha_entrada     | Date              | No necesario          | No                         |
| Hora_entrada      | Character varying | 8 caracteres          | No                         |
| Hora_salida       | Character varying | 8 caracteres          | No                         |
| Horas_sesion      | Character varying | 8 caracteres          | No                         |

**Tabla 8: “ConfigIP”**

Esta tabla es la más pequeña del gestor de base de datos debido a que sólo contiene una columna, es necesario crearla para llevar el control del sistema de reconocimiento y restricción de IP.

La columna es:

- . Estado: Ésta funcionará como bandera (sólo dos estados), sus valores posibles son: “Habilitado” y “Deshabilitado”

La configuración de la columna se muestra enseguida:

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Estado            | Character varying | 13 caracteres         | No                         |

### Tabla 9: “Visitantes”

Es necesario utilizar una tabla para almacenar permanentemente un número creciente, que aumentará cada vez que un usuario nuevo abra la página del SICSS. He creado un algoritmo en PHP para que dicho contador cuente inteligentemente a los usuarios mediante su IP.

Las columnas se muestran a continuación:

- . Id: Es un contador automático que hará saber de forma asíncrona las veces que la página ha sido recargada (lo que haría un contador convencional).
- . Ip: IP real del visitante, la cual servirá para contar el número verdadero de visitantes.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Id                | Serial            | No necesario          | No                         |
| Ip                | Character varying | 15 caracteres         | Si                         |

En el caso de esta tabla se requiere aceptar valores nulos para la IP para evitar que algún usuario experimentado haga que la página tenga errores enviando valores diferentes o nulos a esta tabla.

### Tabla 10: “Mensajes”

En esta última tabla se almacenarán algunos datos y mensajes de los visitantes que deseen dejar alguna opinión de la página, queja o sugerencia. Este pequeño libro de firmas está ubicado en la sección de “Acerca de...”.

Las columnas y configuración se muestran a continuación:

- . Id: Contador de mensajes.
- . Nombre: Nombre del visitante.
- . Correo: Correo electrónico del visitante (esta entrada no es obligatoria).
- . Fecha: Fecha en que se escribió el mensaje.

- . Hora: Hora en que se escribió el mensaje.
- . Ip: Dirección IP del visitante que escribió el mensaje.
- . Mensaje: Mensaje de texto del visitante.

| NOMBRE DE COLUMNA | TIPO DE DATO      | TAMAÑO MÁXIMO DE DATO | ¿SE ACEPTAN VALORES NULOS? |
|-------------------|-------------------|-----------------------|----------------------------|
| Id                | Serial            | No necesario          | No                         |
| Nombre            | Character varying | 25 caracteres         | No                         |
| Correo            | Character varying | 80 caracteres         | Si                         |
| Fecha             | Date              | No necesario          | No                         |
| Hora              | Character varying | 8 caracteres          | No                         |
| Ip                | Character varying | 15 caracteres         | Si                         |
| Mensaje           | Text              | No necesario          | No                         |

La columna “correo” acepta valores nulos debido a que es opcional dejar un correo electrónico, y la columna “Ip” acepta valores nulos por cuestiones de seguridad, mencionadas anteriormente.

#### 4.4.3.3 Autenticación de registro de usuarios

Ahora que ya se tiene el esquema completo de la base de datos es posible interactuar con ella insertando, consultando, eliminando y actualizando registros en cada una de sus tablas. Para realizar una autenticación simple, es decir sin necesidad de emplear protocolos especiales como “https” o encriptación de datos, se requieren hacer algunas comprobaciones mediante consultas para poder insertar los datos en PostgreSQL.

La seguridad en este aspecto no llega a niveles altos, debido a que es muy poca la información que se tratará; imaginando que más de la mitad de los universitarios de la FES Aragón realizará su servicio social dentro del plantel contaríamos con un valor estimado de diez mil registros, además de ser información poco importante para intentar ser corrompida por expertos en seguridad informática, y de cualquier modo, si alguien intentara alterar los datos de registro, se cuenta con la opción de “historial” de la que se

tratará un poco más adelante, que es capaz de almacenar todos los movimientos que se hagan en el sistema.

De éste modo es posible realizar autenticaciones por medio de comparaciones con los valores ya establecidos en la base de datos.

Lo inicial antes de comenzar es definir un archivo que almacene la conexión con la base de datos, será de la siguiente manera:

1. `$Conex="host=localhost port=5432 user=Ricardo`
2. `password=contraseñaAdministrativa dbname=SICSS";`
3. `pg_connect($Conex) or die(pg_error());`

El archivo lo nombraré como "conexionBD.php" donde el usuario y contraseña han sido cambiados por razones de seguridad, así como el nombre del archivo.

Retomando el contenido del archivo "regss.php" expuesto en el capítulo "4.4.2.2 Registrar usuarios" tenemos que la primera interacción con la base de datos PostgreSQL es como sigue:

1. `$queryNick="SELECT * FROM passwords WHERE nick='$nick';`
2. `$sqlNick=pg_query($queryNick) or die(pg_ErrorMessage());`
3. `$numCadsNick=pg_num_rows($sqlNick);`
4. `if ($numCadsNick!=0)`
5. `{`
6. `echo "<h1><br />Error, el nombre de usuario <b>$nick</b> no est&aacute; disponible</h1><br /><br />";`
7. `echo "Por favor selecciona otro";`
8. `}`

Lo primero es realizar una consulta SQL a la tabla "passwords" pidiendo que muestre resultados en donde '\$nick', que es la variable que almacena el nombre de usuario que se introdujo previamente en el formulario de registro, coincida con algún registro (línea 1), si coincide es porque ya existe y por lo tanto se informará al usuario que no es posible utilizar ese nombre de usuario y se detendrá la ejecución del módulo "regss.php", de lo contrario se proseguirá con la autenticación. Del mismo modo se revisará la autenticidad del número de cuenta para comprobar que el usuario no esté registrado previamente, si se introduce un número de cuenta que ya existe se informará al usuario y se detendrá la ejecución de la autenticación, de lo contrario se continuará con la rutina.

Enseguida se debe verificar que las contraseñas que se han introducido en los campos "contraseña" y "confirmar contraseña" sean idénticas, siguiendo el mismo patrón del código antes mostrado.

Si las tres comprobaciones son exitosas se procede con una función de PHP que se encarga de obtener la fecha actual para indicar al usuario el inicio de su servicio social, así como la fecha de término estimada que corresponde a seis meses añadidos a la fecha inicial, contemplando que *el servicio social debe realizarse en un tiempo no menor*

de 6 meses ni mayor de 2 años pero en ningún caso menor de 480 horas<sup>29</sup>, según el artículo 6º del reglamento general del servicio social de la Universidad Nacional Autónoma de México.

Ahora sí, se puede proceder a indicarle a PostgreSQL que la información es correcta y cumple con el formato adecuado para poder almacenarse en su base de datos, esto es mediante una sentencia SQL que se encarga de insertar directamente los datos contenidos en las variables PHP en las tablas sugeridas, por ejemplo si se desean insertar tres variables de PHP en una tabla llamada “valores” con los campos: valor1, valor2 y valor3 y con un formato de campo de “character varying” (los detalles de tipos de dato se detallaron en el capítulo “4.4.3.2.1 Esquema de PostgreSQL”), entonces quedaría del siguiente modo:

1. *pg\_query(“INSERT INTO valores(valor1,valor2,valor3) VALUES (\$variable1’,’\$variable2’,’\$variable3’)”);*

La función “pg\_query” sirve para ejecutar una sentencia SQL dentro de PHP, en este caso es una inserción de datos pero se pueden realizar consultas (SELECT), actualizaciones (UPDATE) o eliminación de registros y tablas (DELETE y DROP).

#### 4.4.3.4 Iniciar conteo

Hasta este momento ya tenemos creado un registro de usuario, ahora hay que comenzar a utilizar el sistema ya que será la acción que concurrirá el prestador de servicio social a lo largo de su estancia en la Facultad.

La primera pantalla que necesitamos estará ubicada en el menú desplegable “Inicio → Iniciar conteo”. Se cuenta hasta el momento con dos alternativas de uso, una es mediante el lector de código de barras antes mencionado, y la otra es utilizando el nombre de usuario que se pidió al momento de registrarse y su contraseña.

Para conseguir esto se emplea la técnica de formularios en HTML para almacenar y enviar los datos que autentifican a un usuario. Sólo se requieren dos campos y un botón de envío, las validaciones son exactamente iguales que las tratadas en el capítulo anterior para lograr comparar los datos con los registrados y de este modo saber si el usuario escribió correctamente sus datos.

Cada vez que se encuentre un error se mostrarán mensajes con la información al usuario para saber a qué se debe, las posibilidades de error son las siguientes:

- Se dejaron algunos campos en blanco.

<sup>29</sup> <http://info4.juridicas.unam.mx/unijus/cmp/leguniv/214.pdf>

- El usuario no existe en la base de datos.
- El usuario ya ha iniciado su conteo de tiempo.
- La contraseña es incorrecta.
- Se produjo un error interno en la base de datos

Existe la posibilidad que el servidor PostgreSQL deje de funcionar, aunque no es nada común que esto suceda pero no se debe dejar de citar la opción, esto puede ocurrir si algún hardware del servidor tiene alguna falla, problemas con la red de la FES Aragón, o simplemente la intervención de algún malware o virus.

El flujo normal de datos en el SICSS es que el usuario introduzca bien su nombre de usuario y contraseña para que el sistema pueda procesar su solicitud. Es aquí donde vuelve a intervenir un nuevo módulo que se encargará de recibir los datos del formulario y validarlos con la base de datos para poder mostrar una pantalla como se muestra a continuación:

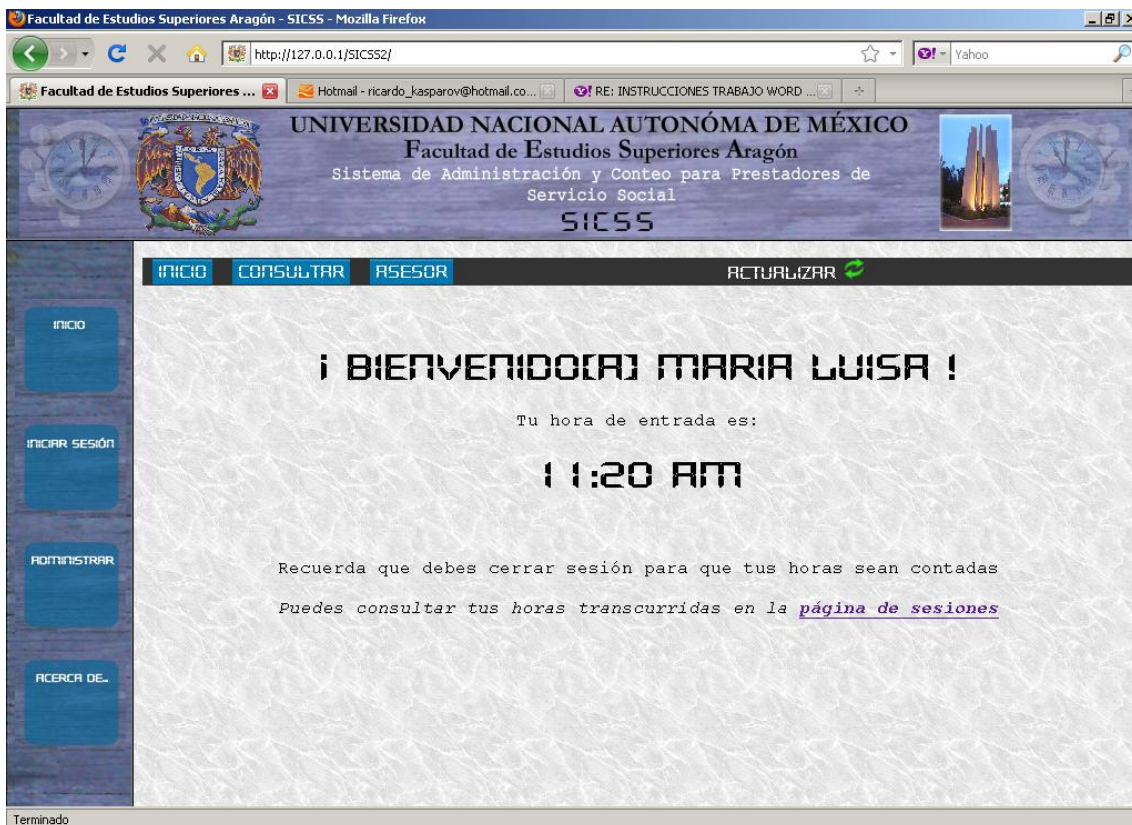


Figura 4.25 - Pantalla de bienvenida al usuario cuando inicia su conteo.

De no ser vista la pantalla anterior entonces debe aparecer en su lugar alguno de los mensajes de error enlistados anteriormente.

El módulo de autenticación del inicio de conteo de un usuario se encarga de crear un registro temporal en PostgreSQL con los datos de registro citados en el capítulo de configuración de la base de datos, para de este modo tener la información necesaria para que al finalizar el conteo de tiempo del usuario se pueda crear un algoritmo que sea capaz de mostrar los datos necesarios para la creación de un historial de registro y la inserción en la tabla que almacena los datos de seguridad para llevar un conteo perfecto del tiempo que labore el prestador de servicio social.

#### 4.4.3.5 Usuarios en línea

Otra opción del menú desplegable es la de consultas, particularmente para esta sección el menú “Consultar → Usuarios activos” que al darle clic direccionará a un módulo que contiene una consulta SQL para visualizar en pantalla una lista con los usuarios que actualmente ha iniciado su conteo de tiempo, y a diferencia de esta misma consulta desde la sección de administración, aquí solo podrán verse los usuarios del lugar o departamento específico desde donde se está realizando la petición.

El funcionamiento se trató en el capítulo “4.4.2.2 Consultas”, con la diferencia ya mencionada que sólo se muestran los usuarios en línea del lugar desde donde se hace la consulta:

The screenshot shows a web browser window with the URL <http://127.0.0.1/SICSS2/>. The page header includes the logo of the Universidad Nacional Autónoma de México and the text: "UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO Facultad de Estudios Superiores Aragón Sistema de Administración y Conteo para Prestadores de Servicio Social SICSS". Below the header is a navigation bar with buttons for "INICIO", "CONSULTAR", "ASESOR", and "ACTUALIZAR". The main content area displays the title "USUARIOS ACTUALMENTE EN FUNAM SALA 5" and a table with the following data:

| No. de Cuenta | Nombre                     | Hora de Entrada | Tiempo transcurrido |
|---------------|----------------------------|-----------------|---------------------|
| 684631634     | RODRIGO GARCÍA RAMIREZ     | 12:36 AM        | 2:27:26             |
| 786578257     | MAYRA MARTINEZ PALACIOS    | 2:16 PM         | 0:47:20             |
| 346458934     | ANGEL MALDONADO ORRUTIA    | 10:35 AM        | 4:28:47             |
| 394857349     | MARIA LUISA RAMIREZ FLORES | 11:36 AM        | 3:27:43             |
| 347862349     | SALVADOR MERINO GONZALES   | 10:05 AM        | 4:58:13             |

Figura 4.26 - Consulta de usuarios activos realizada por asesores.



#### 4.4.3.6 Terminar conteo

Cuando el usuario haya finalizado su labor del día tendrá que indicárselo al sistema, para ello se debe abrir la página contenida en el menú desplegable “Inicio → terminar conteo” en donde se llevara a cabo el mismo procedimiento que cuando se inició el conteo, es decir se deben llenar los campos con el nombre de usuario del prestador de servicio social y su contraseña. Si alguno de los datos es erróneo o se deja en blanco se mostrará el error en pantalla. Las validaciones son las mismas que cuando se inicia el conteo de tiempo.

Como se puede apreciar hay algunas secciones que son muy similares por tal motivo no es necesario repetir todo el procedimiento, validaciones y código, basta con tener presente la función que lleva a cabo cada sección y ubicarla dentro del menú de usuarios.

Cuando se realice este proceso exitosamente se mostrará un mensaje con el tiempo que haya transcurrido y las horas que el usuario lleve acumuladas en total tal y como se muestra a continuación:

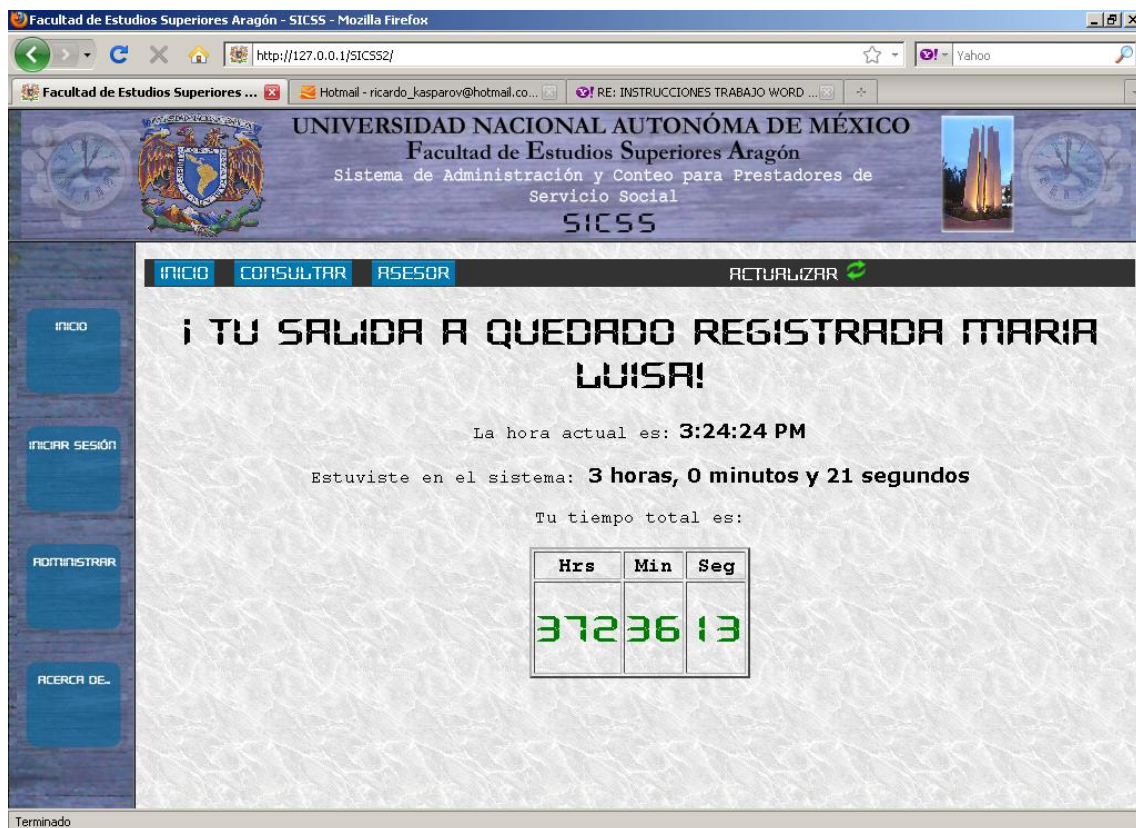


Figura 4.27 - Pantalla de término de conteo del usuario.

Si por algún motivo el usuario olvidó cerrar su sesión existen dos soluciones, una que la próxima vez que entre al sistema cierre su sesión y automáticamente el SICSS agrega el número máximo de horas permitido por día equivalente a 15 horas y los ajustes necesarios los deberá realizar el asesor con ayuda de un administrador; la otra es que el asesor cierre la sesión del usuario en su lugar, para ello se cuenta con la siguiente sección.

#### 4.4.3.7 Forzar baja

Esta opción permite al asesor dar de baja con un solo clic a cualquier usuario que tenga activa su sesión en el sistema, esto es por si algún usuario olvida terminar su conteo, o simplemente por alguna razón el asesor decide terminarlo en algún momento. Para ello se debe entrar al menú desplegable “Asesor → Forzar baja” donde se verá una lista con todos los usuarios que tengan su sesión activa en ese momento y en ese lugar tal y como se muestra a continuación:

The screenshot shows the web interface of the SICSS system. At the top, there is a header for the 'UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO Facultad de Estudios Superiores Aragón Sistema de Administración y Conteo para Prestadores de Servicio Social SICSS'. Below the header is a navigation bar with buttons for 'INICIO', 'CONSULTAR', 'ASESOR', and 'ACTUALIZAR'. The main content area is titled 'USUARIOS ACTUALMENTE EN SISTEMA' and contains a table with the following data:

| No. de Cuenta | Nombre                   | Hora de Entrada | Tiempo transcurrido | Bajas |
|---------------|--------------------------|-----------------|---------------------|-------|
| 346458934     | ANGEL MALDONADO ORRUTIA  | 10:35 AM        | 4:52:06             | Baja  |
| 347862349     | SALVADOR MERINO GONZALES | 10:05 AM        | 4:21:32             | Baja  |
| 684631634     | RODRIGO GARCÍA RAMIREZ   | 12:36 AM        | 2:50:45             | Baja  |
| 786578257     | MAYRA MARTINEZ PALACIOS  | 2:16 PM         | 0:10:39             | Baja  |

Figura 4.28 - Usuarios activos para forzar término de conteo exclusivamente por el asesor.

Esta página está protegida por la contraseña del asesor, eso significa que cada vez que se desee entrar aquí se pedirá escribir la contraseña del asesor.

A un lado de cada registro se agrega un botón con la leyenda “Baja” el cual servirá para terminar el conteo del usuario en cuestión. Aquí no hay lugar a validaciones ya estando dentro de la página, únicamente se preguntará si se está seguro de dicha acción mediante un “aceptar” o “cancelar”, esto por si se da un clic en otro lugar por equivocación.

Cuando se dé clic en “cancelar” sencillamente no ocurrirá nada, si se elige “aceptar” entonces se procederá con la baja del usuario y se mostrará una pantalla de confirmación tal y como se muestra a continuación:

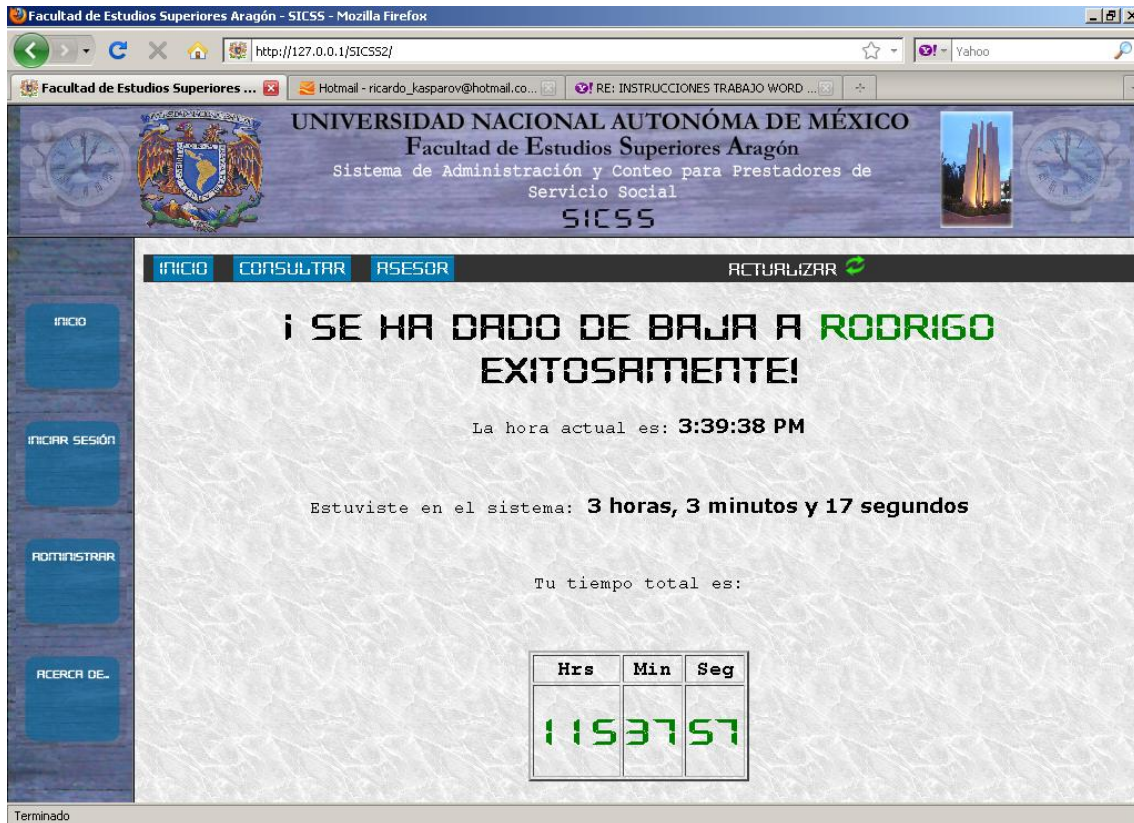


Figura 4.29 - Pantalla consecuente al forzado de término de conteo de tiempo del usuario.

#### 4.4.4 SECCIÓN DE “ACERCA DE”

Finalmente he introducido un apartado especial para quienes deseen conocer un poco más de los temas que me ayudaron a desarrollar el sistema SICSS, así como algunos agradecimientos, el contacto con el área administrativa y un pequeño espacio para dejar algún comentario.

Al dar clic sobre el botón “Acerca de...” enviará a un módulo PHP que obtiene información de la base de datos para mostrar los comentarios que se han dejado, así mismo hice notar los links recomendados y el contacto con una atractiva nube de etiquetas, desarrollada con AJAX y montada en un archivo de Adobe® Flash® de tipo SWF.

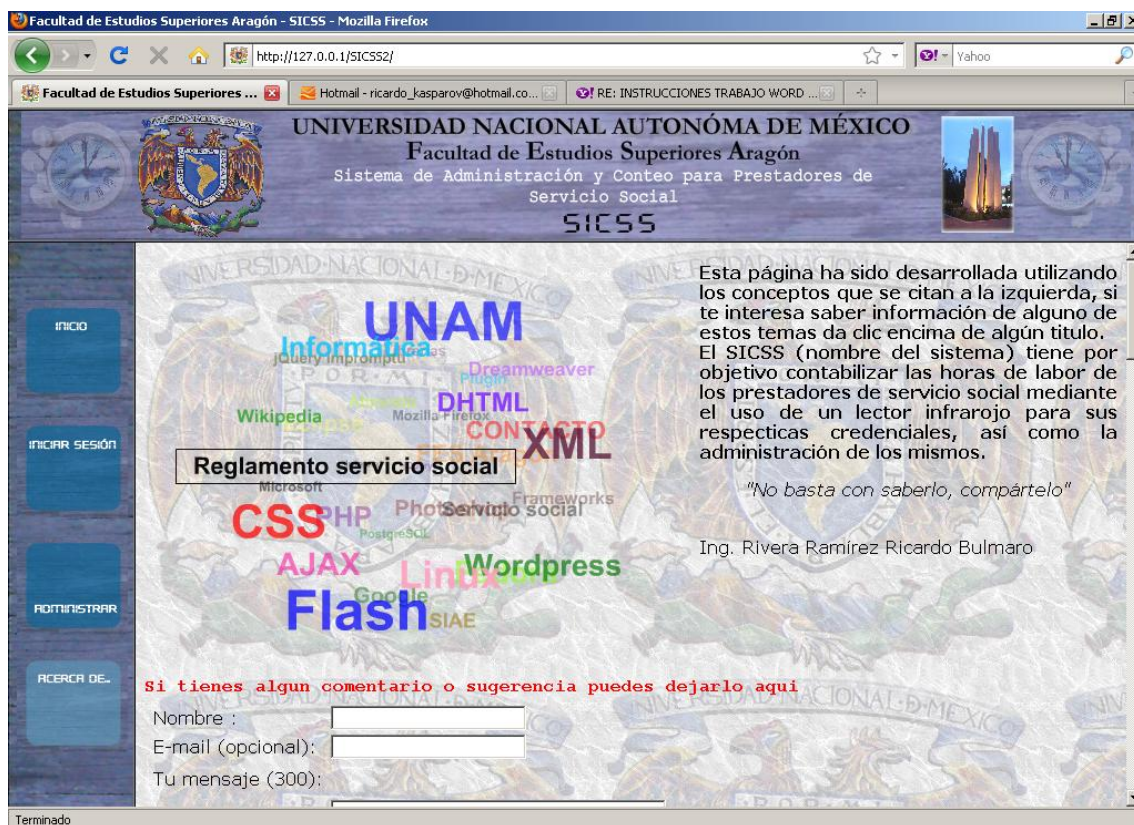


Figura 4.30 - Sección para conocer más sobre el desarrollo de la página SICSS.

Cada vez que se elija alguna dirección en la nube de etiquetas con el título de interés del lector, se desplegará en la sección del “main-frame” el contenido de la página. Si por algún motivo algún cibernauta se interesa por cualquier código empleado en este proyecto, se cuenta con el vínculo etiquetado como “Contacto” en donde se desplegarán los correos disponibles a donde podrá enviar dicha petición.



Figura 4.31 - Pantalla con datos para contactarnos, ubicada en nube de etiquetas.

Por otro lado se ha incorporado un pequeño blog de comentarios en donde se podrán dejar algunas sugerencias o comentarios, mismos que podrán ser vistos por cualquier persona de la red.

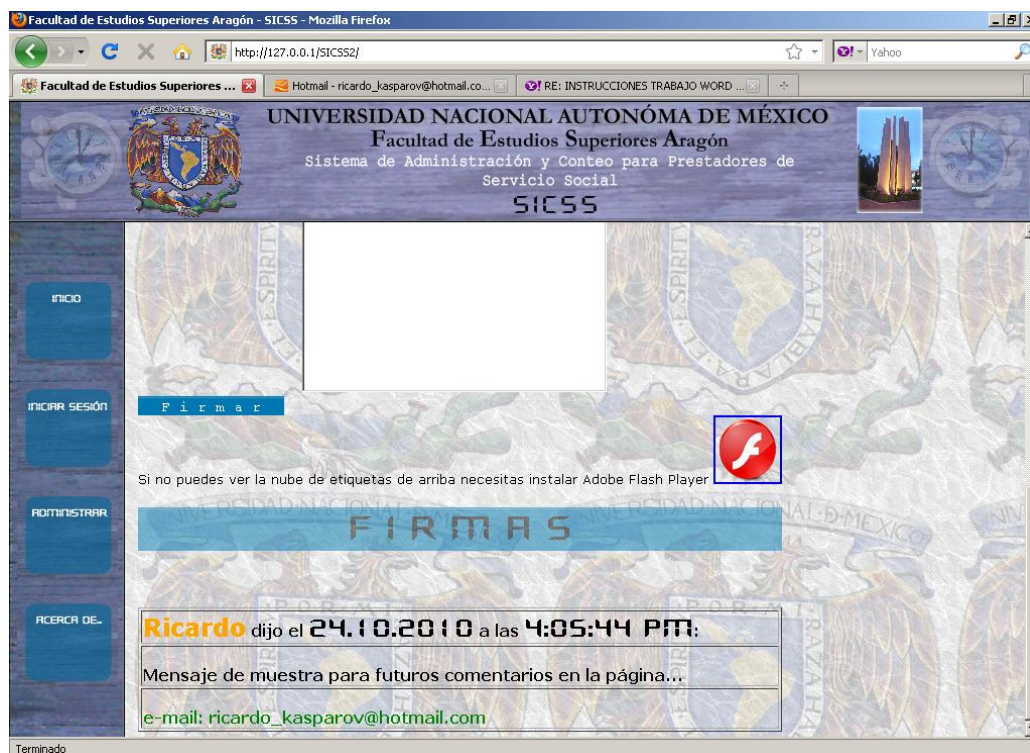


Figura 4.32 - Sección de comentarios o firmas que se deseen dejar en la página.

El mantenimiento del SICSS consta de las siguientes tareas:

- Verificar que el hardware de almacenamiento que aloja a la base de datos tenga espacio suficiente, mínimo 10 Gb, de no ser así liberar espacio en disco duro.
- Crear respaldos de la base de datos con frecuencia (una vez por semana).
- Realizar mantenimiento preventivo y correctivo de servidores que contengan al sistema (limpieza física y eliminación de archivos temporales).
- Verificar coherencias en los historiales por posibles fraudes.
- Observar comentarios realizados en la página por si alguien informa de un error en ella.
- Mejorar módulos del SICSS para mayor agilidad de procedimientos, implementación de nuevas herramientas y corrección de futuros errores.

Esta tarea podrá realizarse por cualquier estudiante de la FES Aragón que realice su servicio social en el departamento de informática, y sea autorizado por el administrador del mismo en ese momento, hago mención de esto ya que todo el código se encuentra comentado y es muy legible para programadores novatos, la distribución o mal uso del mismo será responsabilidad del encargado del departamento en cuestión.

# CONCLUSIONES

---

Para exponer por completo el desarrollo de una página web y que sea de sencilla comprensión a cualquier lector, se requeriría de un libro mucho más extenso que una tesis debido al amplio contenido de temas computacionales. Consideré adecuado exponer un panorama de la base que sostiene a este proyecto, me refiero a los principios de la computación, sistemas básicos, automatización de procesos y funcionalidades de los sistemas web en la actualidad.

Se pudo comprobar que un sistema bien planificado es confiable, eficaz, veloz y tan innovador como lo deseas, así como poder finalmente utilizarlo y con solo algunos clics de mouse poder manejar toda la estructura que está por detrás de la página Web.

Puedo decir que los objetivos, en modo piloto, han sido alcanzados debido a que el sistema aún no está en producción, sin embargo en pruebas que he realizado al sistema, funcionalmente no tiene problemas y algunas personas que son ajenas a la informática que han entrado al portal lo han encontrado bastante sencillo de utilizar, lo cual me hace cubrir uno de mis objetivos principales.

Para conseguir cubrir totalmente los objetivos planteados en este proyecto, será necesario llevar a cabo una difusión en la universidad de tal herramienta de trabajo, así como gente del área de informática que esté dispuesta a administrar el sistema, así como checar cualquier anomalía en el mismo por pérdidas de energía u otros factores, como la falla de algún dispositivo hardware.

En general debo reconocer que existen proyectos de sistemas en internet que son capaces de realizar tareas mucho más complejas y grandes que el SICSS, pero eso no implica que no me halla costado más de un año de trabajo este proyecto, apoyándome en libros y algunos cursos que he tomado, claro sin dejar de lado el aprendizaje que me formó a lo largo de 5 años durante mi licenciatura, por lo que me enorgullece poder hacer un pequeño aporte a lo que es y siempre será mi segundo hogar, la Universidad Nacional Autónoma de México.

# REFERENCIAS

---

## Bibliografía

- H. Sampieri Roberto, "Metodología de la investigación", Mc Graw-hill, segunda edición, Mexico 2000.
- Floyd Thomas L., "Fundamentos de sistemas digitales", Pearson Educación, Madrid 2000.
- Leon Presser, "Ciencias de la computación", Limusa, México 1979.
- Aréchiga Rafael, "Fundamentos de computación", Limusa, México 1978.
- Bishop Peter, "Computadores de la quinta generación", Paraninfo, Madrid 1989.
- Bartee Thomas C., "Fundamentos de computadoras digitales", McGraw-Hill, USA 1981.
- Sherman Philip M., "Técnicas de programación de computadores", Prentice Hall, USA 1970.
- Fournier Julio, "Introducción a la informática", Limusa, México 1993.
- Ferreira Cortès Gonzalo, "Informática para cursos de Bachillerato, Alfaomega
- Lorenzo Francisco, "Introducción a la computación a través de logo", Limusa, México 1985.
- Sanders Donald H., "Informática: presente y futuro", Mc Graw-hill, México 1994.
- Royo Javier, "Diseño Digital", Paidós Ibérica, 2004.
- Ferreyra Cortés Gonzalo, "Informática 1, la herramienta del nuevo milenio", Alfaomega, México, 2000.
- Revista - Reparación y ensamblado de computadoras PC. Lección 1. Edición Computación aplicada. México 2003.
- Stoutemyer David R., "Programación para ciencia e Ingeniería", Prentice Hall, España 1972.
- Molina Antonio, "Metodología y tecnología de la programación", Repproval, Valencia 1997.
- Holzschlag Molly E., "La biblia de HTML 4", Anaya, Madrid 2000.
- Schwarte Joachim, "El gran libro de HTML - Cómo publicar en internet", marcombo, Barcelona 1996.
- Bobadilla S. Jesús, "HTML dinámico, ASP y JavaScript a través de ejemplos", RA-MA, España 1999.
- Gutiérrez R. Abraham, "PHP 5 a través de ejemplos", Alfaomega, México 2005.
- Fowler Martin, "UML Gota a Gota", Pearson, México 1999.



## **Páginas Web**

- [http://www.librosweb.es/css\\_avanzado](http://www.librosweb.es/css_avanzado)
- <http://es.wikipedia.org>
- [http://oreilly.com/pub/a/oreilly/news/languageposter\\_0504.html](http://oreilly.com/pub/a/oreilly/news/languageposter_0504.html)
- <http://www.computerworld.com/databasetopics/data/software/story/0,10810,10972,2,00.html>

# GLOSARIO

---

---

**ALGORITMO:** Conjunto de instrucciones que permite la resolución de un problema paso a paso.

**BIT:**(dígito binario ) un dígito simple de un numero binario (1 ó 0) en el computador.

**BYTE:** Grupo de bits adyacentes operados como una unidad, (grupos de 8 bits representan un carácter, por ejemplo: a, 7, \$, “, etc.).

**DIAGRAMA DE FLUJO:** Es la representación gráfica de una secuencia de instrucciones de un programa que ejecuta un computador para obtener un resultado determinado.

**CAPTURISTA:** En informática, un capturista es una persona encargada de obtener e introducir datos en bases de datos que serán utilizadas por usuarios finales o introducir esos mismos datos en las estructuras de datos internas de los programas que serán utilizados por usuarios finales.

**CÓDIGO FUENTE:** Programa en su forma original, tal y como fue escrito por el programador, el código fuente no es ejecutable directamente por el computador, debe convertirse en lenguaje de máquina mediante compiladores, ensambladores o interpretes.

**COMPILADOR:** Programa de computadora que produce un programa en lenguaje de máquina, de un programa fuente que generalmente esta escrito por el programador en un lenguaje de alto nivel.

**CONSTANTE:** Valor o conjunto de caracteres que permanecen invariables durante la ejecución del programa.

**FRAMES:** Los marcos o frames permiten que se pueda mostrar varias páginas con distintas URLs (direcciones web) en una sola ventana y que al dar clic en un enlace solo cambie parte de la página sin alterar el resto.

**HARDWARE:** Es la parte tangible del computador.

**INTERPRETE:** Dispositivo o programa que recibe una por una las sentencias de un programa fuente, la analiza y la convierte en lenguaje de máquina si no hay errores en ella. También se puede producir el listado de las instrucciones del programa.

**LOGIN:** En el ámbito de seguridad informática, login o logon (en español ingresar o entrar) es el proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.

**MNEMOTÉCNICOS** - La nemotecnia o mnemotecnia es el procedimiento de asociación mental de ideas, esquemas, ejercicios sistemáticos, repeticiones, etc. para facilitar el recuerdo de algo.

**PLUGIN:** Un plugin o complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

**PROGRAMA (APLICACIÓN):** Es un conjunto de instrucciones que una vez ejecutadas realizarán una o varias tareas en una computadora.

**SOFTWARE:** Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible del computador.

**VARIABLE:** En programación es una estructura que contiene datos y recibe un nombre único dado por el programador, mantiene los datos asignados a ella hasta que un nuevo valor se le asigne o hasta que el programa termine.

**VÍNCULO:** Un hipervínculo (también llamado enlace, vínculo, o hiperenlace) es un elemento de un documento electrónico que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento, o bien, a alguna página Web.