



INSTITUTO SUPERIOR TECNOLÓGICO
SUDAMERICANO
QUITO - ECUADOR

ESCUELA DE
SISTEMAS DE AUTOMATIZACIÓN

PROYECTO DE TITULACIÓN

TEMA:

**Diseño e implementación de un Sistema para el Registro de
Asistencia de los estudiantes del Instituto Superior
Tecnológico Sudamericano Quito
(RAE-SUDA)**

AUTORES: Quishpe Andrade Erick Xavier
Rojas Jiménez Junior Fabricio

TUTOR: MSc. FABRIZIO VILLASÍS

San Francisco de Quito, noviembre del 2019

AUTORÍA

Yo, Erick Xavier Quishpe Andrade, portador de la cédula de ciudadanía No.172168961-8, declaro bajo juramento que el trabajo aquí descrito, es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado e investigado en base a las referencias bibliográficas que se incluyen en este documento. Esta investigación no contiene plagio alguno y es resultado de un trabajo serio desarrollado en su totalidad por mi persona.

Erick Xavier Quishpe Andrade

AUTORÍA

Yo, Junior Fabricio Rojas Jiménez, portador de la cédula de ciudadanía No.172650848-2, declaro bajo juramento que el trabajo aquí descrito, es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado e investigado en base a las referencias bibliográficas que se incluyen en este documento. Esta investigación no contiene plagio alguno y es resultado de un trabajo serio desarrollado en su totalidad por mi persona.

Junior Fabricio Rojas Jiménez

CERTIFICACIÓN

Una vez que se ha culminado la elaboración del proyecto de titulación cuyo tema es: “**Diseño e implementación de un Sistema para el Registro de Asistencia de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito (RAE-SUDA)**”, certifico que el mismo se encuentra habilitado para su defensa pública.

MSc. Fabrizio Villasís
Coordinador de la Escuela de
Sistemas de Automatización
Instituto Superior Tecnológico Sudamericano Quito

CERTIFICACIÓN

Certifico que el señor Erick Xavier Quishpe Andrade ha realizado y concluido su trabajo de titulación, cuyo tema es: “Diseño e implementación de un Sistema para el Registro de Asistencia de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito (RAE-SUDA)”, para obtener el título de Tecnólogo en Sistemas de Automatización, bajo mi tutoría.

MSc. Fabrizio Vicente Villasís Chiriboga
Director del Proyecto de Titulación

CERTIFICACIÓN

Certifico que el señor Junior Fabricio Rojas Jiménez ha realizado y concluido su trabajo de titulación, cuyo tema es: “Diseño e implementación de un Sistema para el Registro de Asistencia de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito (RAE-SUDA)”, para obtener el título de Tecnólogo en Sistemas de Automatización, bajo mi tutoría.

MSc. Fabrizio Vicente Villasís Chiriboga
Director del Proyecto de Titulación

AGRADECIMIENTOS

Expreso mi agradecimiento principalmente al Instituto Superior Tecnológico Sudamericano por la oportunidad que me otorga en el anhelado fin de conseguir mi objetivo de vida en superación. A la Escuela de Sistemas de Automatización por permitirme el desarrollo tanto personal como profesional.

Al personal docente por su esmero y discernimiento durante todo el lapso de aprendizaje el cual no ha sido en vano ya que implemento en nuestra formación ha sido muy satisfactorio.

Finalmente quiero expresar mi más grande y sincero agradecimiento al MSc. Fabrizio Villasís, principal colaborador durante todo este proceso, quien con su conocimiento, enseñanza y paciencia me permitió el desarrollo de este trabajo.

Erick Quishpe

AGRADECIMIENTOS

Me gustaría agradecer en estas líneas la ayuda que muchas personas y colegas me han prestado durante el proceso de creación y redacción de este proyecto. En primer lugar, quisiera agradecer a mis padres que me han ayudado y apoyado en todo mi proceso estudiantil, a mi tutor, Fabrizio Villasís, por haberme orientado en todos los momentos que necesité sus consejos.

Así mismo, deseo expresar mi reconocimiento al Instituto Superior Tecnológico Sudamericano Quito por abrirme sus puertas y permitirme ser parte de tan prestigiosa Institución, por haberme brindado el conocimiento necesario para realizar este proyecto.

A todos mis amigos, vecinos y futuros colegas que me ayudaron de una manera desinteresada, gracias infinitamente por toda su ayuda y buena voluntad.

Junior Rojas

DEDICATORIA

Esta tesis está dedicada para Dios por bendecirnos la vida, por guiarnos a lo largo de nuestra existencia, por ser el apoyo y la fortaleza en aquellos momentos de dificultad y de debilidad.

A mis padres, quienes me enseñaron que la mejor herramienta que se puede tener para alcanzar nuestros sueños es el conocimiento y sobre todo la preparación profesional, que incluso la tarea más grande se puede lograr si se hace un paso a la vez.

Erick Quishpe

DEDICATORIA

Esta tesis está dedicada a mi familia por haber sido mi apoyo a lo largo de toda mi carrera tecnológica y a lo largo de mi vida. A todas las personas que supieron acompañarme en esta etapa de mi existencia, aportando grandes consejos y ayudándome a mi formación profesional y como ser humano.

Junior Rojas

RESUMEN

Se trata de un sistema llamado Registro de Asistencia de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito (RAE-SUDA), el cual fue desarrollado mediante software y hardware libre, es una plataforma fácil de programar y de manejar.

El RAE-SUDA contiene una placa llamada Arduino UNO, una Shield Ethernet W5100, que es capaz de conectarse a una red montada sobre un Arduino UNO y conectado a un Módulo RC-522 el cual es capaz de leer tarjetas magnéticas y enviar esta información a una Base de Datos en tiempo real.

Para poder visualizar la información enviada a la Base de Datos se creó una página HTML capaz de mostrar los datos que se enviaron e información de los estudiantes que previamente se cargó en esta Base de Datos y cuando sea necesaria esta información poderla descargar en formato PDF y XML desde la página HTML.

Con este sistema y Base de Datos se soluciona el problema de registro de asistencia en hojas, pérdida de registros de asistencia, molestias de los profesores para llevar registros en hojas, verificación de asistencia de manera más fácil y optimización de recursos institucionales.

ABSTRACT

It is a system called Student Attendance Registry of the Quito South American Institute of Technology (RAE-SUDA), which was developed using free software and hardware, it is an easy platform to program and manage.

The RAE-SUDA contains an Ethernet Shield that is capable of connecting to a network mounted on an Arduino UNO and connected to an RC-522 Module which is capable of reading magnetic cards and sending this information to a database in real time.

In order to visualize the information sent to the database, an HTML page was created capable of displaying the data that was sent and student information that was previously loaded in this database and when necessary this information can be downloaded in PDF format and XML from the HTML page.

With this system and Database the problem of registration of attendance in sheets, loss of attendance records, inconvenience of teachers to keep records in sheets, verification of assistance in an easier way and optimization of institutional resources is solved.

CONTENIDO

1.	Introducción.....	1
2.	Justificación.....	3
3.	Antecedentes	4
4.	Objetivos	5
4.1	Objetivo General	5
4.2	Objetivos Específicos	5
5.	Marco Teórico	6
5.1	Microcontroladores.....	6
5.2	Arduino.....	7
5.2.1	Ethernet Shield.....	8
5.3	Sistemas de identificación por radio frecuencia(RFID).....	9
5.3.1	Elementos de un Sistema RFID	11
5.3.2	Principios básicos de funcionamiento de un sistema RDID.....	12
5.3.3	¿Cómo funcionan los RFID Tags?	13
5.3.4	Evolución de los sistemas RFID	13
5.3.5	Módulo RFID RC-522.....	14
5.3.5.1	Comunicación SPI.....	14
5.4	Bases de Datos	15
5.4.1	Herramientas para crear Bases de Datos.....	16
5.5	jQuery.....	17
5.6	EXCEL y su manejo de datos.....	18
6.	Desarrollo del proyecto de titulación.....	19
6.1	Requerimientos del sistema y tecnologías disponibles... ..	19
6.2	Diseño del Sistema	20
6.3	Implementación del sistema	21
6.3.1	Creación de la Base de Datos y su interfaz gráfica	26
6.3.2	Creación de filtro de asistencia en EXCEL.....	53
6.4	Pruebas de correcto funcionamiento	55
7.	Conclusiones y Recomendaciones... ..	58
7.1	Conclusiones	58
7.2	Recomendaciones.....	58
	Referencias... ..	59

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de un microcontrolador PIC	6
Figura 2. Logo de Arduino que muestra que es un software libre	7
Figura 3. Placa Arduino	7
Figura 4. Partes del Arduino Uno	8
Figura 5. Ethernet Shield	9
Figura 6. Sistema básico RFID	9
Figura 7. Tag pasivo modelo tarjeta	10
Figura 8. Tag RFID activo modelo SYTAG245-2s	11
Figura 9. Elementos de un sistema RFID	12
Figura 10. Módulo RC-522 para Arduino	14
Figura 11. Conexión SPI	15
Figura 12. Pines de conexión del Módulo RC-522 a Arduino	15
Figura 13. Funcionamiento de una Base de Datos	16
Figura 14. JQuery muestra que es fácil de programar con JavaScript	18
Figura 15. Diagrama de funcionamiento del sistema de registro de asistencia	20
Figura 16. Esquema de diseño del Sistema	20
Figura 17. Dispositivo hardware de envío de información a la Base de Datos	21
Figura 18. IDE para programar e interactuar con el hardware de Arduino	22
Figura 19. Configuración de pines y librerías necesarias	22
Figura 20. Configuración de Ethernet Shield y comprobación de conexión a la red administrativa	23
Figura 21. Lectura de tarjetas	24
Figura 22. Conexión a la Base de Datos y envío de identificación de tarjeta	24
Figura 23. Archivo de conexión con la Base de datos	25
Figura 24. Archivo para guardar la identificación de la tarjeta en la Base de Datos	25
Figura 25. Funcionamiento de la Base de Datos con la interfaz gráfica	26
Figura 26. Creación de la Base llamada registro asistencia alumnos	26
Figura 27. Creación de la tabla llamada data con sus campos necesarios	27
Figura 28. Creación de la tabla llamada estudiantes con sus campos necesarios	28
Figura 29. Pantalla de ejemplo de información de la tabla data	28
Figura 30. Pantalla de ejemplo información de la tabla estudiantes	29
Figura 31. Relación entre la tabla data y estudiantes	29

Figura 32. Agregado de librerías para datatable y título de la página HTML	30
Figura 33. Creación de la tabla dt_cliente y sus columnas con sus respectivos nombres	30
Figura 34. Agregado de librerías para exportar datos a Excel y PDF, lectura de los datos	31
Figura 35. Creación de la función listar	32
Figura 36. Creación de botones para exportar a Excel y PDF	32
Figura 37. Creación de la función idioma_espanol ()	33
Figura 38. Creación de seguridad para la página	33
Figura 39. Programación del archivo conexión.php para poder conectar la página de asistencia de alumnos a la base de datos	34
Figura 40. Programación del archivo listar.php para poder mostrarlos datos dentro de la base de datos en la página asistencia de alumnos	34
Figura 41. Funcionamiento de la página de asistencia donde se muestra un ejemplo de asistencia	35
Figura 42. Se agregan las librerías necesarias	36
Figura 43. Formulario de registro de nuevos estudiantes	37
Figura 44. Creación de datatable y nombres de las columnas	37
Figura 45. Formulario para eliminación de estudiantes	38
Figura 46. Declaración de librerías para botones de datatable y lectura de funciones listar (), guardar (), eliminar ()	39
Figura 47. Creación de la función guardar ()	40
Figura 48. Creación de la función eliminar ()	40
Figura 49. Creación de la función mostrar_mensaje ()	41
Figura 50. Creación de la función listar ()	41
Figura 51. Elección de idioma español de la tabla, botón de agregar usuarios nuevos	42
Figura 52. Creación de las funciones agregar_nuevo_usuario (), obtener_data_editar (), obtener_id_eliminar ()	43
Figura 53. Creación de la función idioma_espanol ()	43
Figura 54. Creación del archivo conexión.php de la página registro de alumnos	44
Figura 55. Programación del archivo listar.php de la página registro de alumnos para poder mostrar los alumnos registrados dentro de la Base de Datos	44
Figura 56. Creación de switch para elección si se desea registrar, modificar o eliminar alumnos	45
Figura 57. Creación de funciones existe_usuario, registrar y modificar	46
Figura 58. Creación de funciones eliminar, verificar_resultado y cerrar	46

Figura 59. Funcionamiento de la página de registro de alumnos a la Base de Datos.....	47
Figura 60. Formulario para el registro de nuevos estudiantes.....	48
Figura 61. Cambio de datos de un estudiante.....	48
Figura 62. Personal que interactúa con la Base de Datos.....	48
Figura 63. Página de inicio del Sistema de Registro de Asistencia.....	49
Figura 64. Página login.html.....	49
Figura 65. Menú que muestra la asistencia por aulas y registro de estudiantes nuevos.....	50
Figura 66. Menú que muestra la asistencia por aulas y registro de estudiantes nuevos.....	51
Figura 67. Archivo de conexión hacia la Base de Datos de los usuarios.....	51
Figura 68. Creación de la seguridad para el archivo de redirección.php.....	52
Figura 69. Creación del del menú de Aulas y Registro de Estudiantes.....	52
Figura 70. Filtro de asistencia de Estudiantes en EXCEL.....	53
Figura 71. Programación para filtro de datos en EXCEL.....	53
Figura 72. Consulta hacia archivo EXCEL que contiene la Base de Datos de la asistencia....	54
Figura 73. Tabla comparativa de pruebas realizadas.....	55
Figura 74. Funcionamiento y ubicación del servidor en el 4to piso del edificio matriz de la institución.....	55
Figura 75. Prueba de registro de asistencia.....	56
Figura 76. Error de conexión hacia la red.....	56
Figura 77. Error de conexión hacia la Base de Datos.....	57
Figura 78. Funcionamiento del sistema en el edificio matriz de la Institución.....	57

LISTA DE ANEXOS

ANEXO 1: ARDUINO UNO DATASHEET

ANEXO 2: ETHERNET SHIELD DATASHEET

ANEXO 3: RC-522 DATASHEET

ANEXO 4: AGREGAR NUEVOS ESTUDIANTES Y ASOCIAR IDENTIFICACIÓN DE
TARJETA

1. Introducción

En el presente proyecto se plantea realizar el diseño e implementación de un Sistema de Registro de Asistencia a Clases de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito, pero particularmente para el tercero y cuarto piso de las instalaciones del edificio Matriz ubicado en la av.10 de agosto N35-108 e Ignacio San María.

El Sistema de Registro de Asistencia a clases se plantea para modernizar el registro de asistencia en papel al pasar a un sistema informático, el cual registrará la fecha y la hora, el día de la semana, el número de identificación de la tarjeta, los nombres y apellidos del estudiante, y la Carrera que estudia. El tener un registro de asistencia automatizado de asistencia de los estudiantes facilita, mejora y optimiza los recursos institucionales al momento de que cualquier docente o personal administrativo necesite información de la asistencia a clases de cualquier estudiante de una manera mucho más rápida. También podrá obtenerla en formato digital PDF o EXCEL de acuerdo a la necesidad que se tuviese.

El Sistema de Registro de Asistencia contará con una interfaz gráfica sencilla y fácil de utilizar, que permita la consulta de asistencias a clases por parte del docente de una manera intuitiva; en la cual el profesor ingresa a la página de inicio de la interfaz gráfica, ahí le solicitará un Usuario y Contraseña proporcionados por el administrador del sistema, dentro de la siguiente página se encontrarán 5 pestañas en las cuales están el registro de nuevos estudiantes y la asistencia de estudiantes de las aulas 301, 401, 402 y 403, ahí se selecciona el aula de la cual se quiera obtener la asistencia. El sistema estará en un servidor en el que se encontrará alojada la interfaz, ubicada en el cuarto piso debido a que es una Base de Datos que contiene información y no se la puede alterar.

Con este sistema se busca modernizar y dar una solución a la inseguridad de tener un registro de asistencia manual en el cual los estudiantes tienen que firmar hojas y evitar problemas como una firma en un día que pudiera no haber ido a clases o que el docente tome la asistencia verbalmente y que algún estudiante respondiera a ella sin ser la persona en cuestión, para luego proseguir a registrar esto en papel. Por otro lado, se evitaría el uso de papel ya que no se lo necesitaría en este sistema y si por algún motivo un estudiante quiere verificar que su asistencia esté correcta lo puede hacer digitalmente.

El contenido del presente documento contiene la siguiente estructura: como primera parte se hace un estudio del problema del control de asistencia actual en el Instituto Superior Tecnológico Sudamericano Quito. Luego se hace referencia al diseño del sistema para el control de asistencia de los alumnos. Posteriormente una comparativa del tipo de tecnología disponible en el mercado local para poder realizar este sistema, para luego pasar a la elaboración e implementación del Sistema de Control de Asistencia en las aulas del tercer y cuarto piso de las instalaciones del Instituto, pero dejando a disposición de este la futura implementación en los pisos y aulas que se consideren necesarios.

Por último, realizar las pruebas de funcionamiento para finalmente presentar las conclusiones y recomendaciones.

2. Justificación

Actualmente el control de registro de asistencia se realiza mediante firmas en hojas impresas siendo este un método arcaico, desperdiciando tiempo, espacio y material. De darse la necesidad de revisar la asistencia de un estudiante, en un determinado día, esto implicaría revisar el archivo de registros de asistencias por firmas, hoja por hoja, creando con esto ineficiencia y malestar tanto para el estudiante como para el profesor. De aquí la necesidad de un sistema digital que permita automatizar, unificar, y llevar un historial de las asistencias en el transcurso de todo el semestre. Con esto se beneficiará a la Institución al tener un sistema propio y original creado por sus propios estudiantes para el control y registro de asistencia.

3. Antecedentes

El proyecto Sistema de Registro de Asistencia (RAE) surge de la necesidad de mejorar el control de asistencia de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito, ya que este registro se lo realiza mediante hojas en las que el profesor toma lista a los estudiantes que han asistido. Otro método de registro es en el que los estudiantes firman en una hoja, por día de clase.

Algunos profesores no siempre llevan con ellos los registros de asistencia ya que resultan incómodos y molestos.

El Instituto actualmente no cuenta con una Base de Datos y un sistema que registre la asistencia de los estudiantes de manera eficiente y confiable, cada semestre.

El problema al no contar con una Base de Datos de registro de asistencia es que el profesor no puede dirigirse a alguna computadora y poder revisar el registro de asistencia de un estudiante en específico, de una manera fácil, sin pérdida de tiempo y de forma más eficiente; también el desperdicio de hojas en cada semestre ya que estas hojas solo sirven una vez, después hay que desecharlas luego de uno o dos semestres.

Además, está el problema de que si algún profesor pierde estas hojas de registro no puede recuperar el dato de asistencia de ninguna manera, mientras que teniendo este registro en una Base de Datos se puede revisar cuantas veces sea necesario sin temor a perder el valioso registro de asistencia de los estudiantes.

En resumen, al no contar con un sistema de registro de asistencia automatizado resulta una pérdida de tiempo y de recursos, con el riesgo de la eliminación y/o alteración de los registros de asistencia de los estudiantes a una hora de clase.

Es por todas estas razones que el presente proyecto tiene como objetivo diseñar, desarrollar un sistema que registre la asistencia en una Base de Datos de los estudiantes del Instituto Tecnológico Sudamericano Quito a fin de mejorar el registro de asistencia, facilitar la consulta y la optimización de recursos del mismo.

4. Objetivos

4.1 Objetivo General

Diseñar e implementar un sistema para el registro de asistencia a clases de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito, en el tercer y cuarto piso del edificio Matriz en los laboratorios de Informática 301, 401, 402 y 403.

4.2 Objetivos Específicos

1. Definir los requerimientos del sistema e investigar las tecnologías disponibles.
2. Diseñar el sistema de registro de asistencia.
3. Implementar el hardware necesario para el sistema.
4. Desarrollar una Base de Datos que contenga el registro de asistencia de los estudiantes con interfaz gráfica.
5. Implementar el sistema en los laboratorios de Informática (aulas 301, 401, 402 y 403) del edificio Matriz del Instituto Superior Tecnológico Sudamericano Quito.
6. Realizar las pruebas de funcionamiento necesarias.

5. Marco Teórico

5.1 Microcontroladores

Un microcontrolador es un circuito integrado o chip que en su interior contiene las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de Entrada/Salida.

Es capaz de realizar múltiples tareas gracias a que es programable a las diversas necesidades de cada persona que vaya a realizar cualquier tipo de trabajo, antes de trabajar con cualquier microcontrolador es necesario crear datos en un lenguaje para microcontrolador y luego grabarlo en su memoria, estos datos se guardan en formato del sistema numérico hexadecimal que es el sistema con el que trabaja el microcontrolador cuando este es alimentado con una fuente de energía. [1]



Figura 1. Ejemplo de un microcontrolador PIC.

Fuente: <https://es.wikipedia.org/wiki/Microcontrolador>

5.2 Arduino



Figura 2. Logo de Arduino que muestra que es un software libre.

Fuente : <https://www.arduino.cc/en/Trademark/CommunityLogo>

Arduino es una plataforma de software y hardware libre capaz de desarrollar cualquier tipo de proyecto basado en electrónica, cuenta con un lenguaje propio de Arduino.

Existen diferentes tipos de placas, pero la más utilizada es la UNO que cuenta con un microcontrolador ATmega328, tiene Pines digitales que pueden configurarse como entrada (para leer sensores) o como salida (para controlar actuadores), pines analógicos de entrada que usan un convertor analógico/digital (PWM), también tiene puertos de comunicación: USB, serie, I2C y SPI. [2]



Figura 3. Placa Arduino Uno.

Fuente: <https://store.arduino.cc/usa/arduino-uno-rev3>

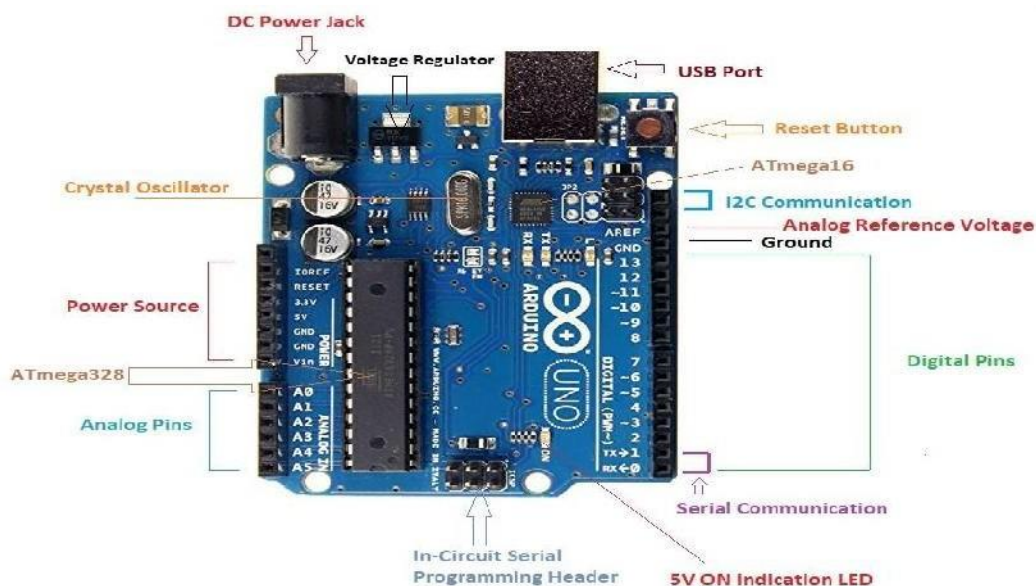


Figura 4. Partes del Arduino Uno.

Fuente: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html>

5.2.1 Ethernet Shield

La Ethernet Shield permite a una placa Arduino conectarse a internet. La Shield está basada en el chip Ethernet Wiznet W5100 el cual provee de una pila de red IP capaz de TCP y UDP soporta hasta cuatro conexiones de sockets simultáneas usa la librería Ethernet para escribir programas que se conecten a Internet usando la Shield, es compatible con el Arduino UNO y Arduino Mega, el Shield nos proporciona un conector Ethernet estándar RJ45 y un conector lector de tarjeta Micro SD, el botón de reset en la Shield resetea el chip W5100 y la placa Arduino, la Ethernet Shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino. [8]

Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la Ethernet Shield. Estos pines no pueden ser usados para e/s genéricas. [8]

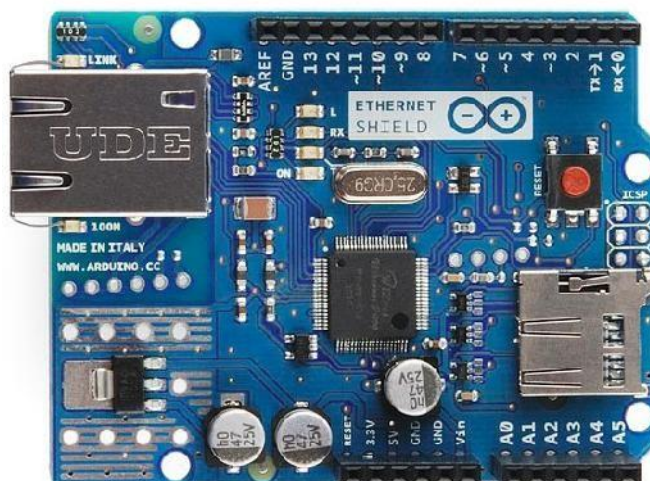


Figura 5. Ethernet Shield.

Fuente: <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>

5.3 Sistemas de identificación por radio frecuencia (RFID)

La identificación por Radio Frecuencia (RFID) es un sistema en el cual leemos, escribimos datos en dispositivos denominados etiquetas, tarjetas o transpondedores RFID.

La creación de estos Sistemas de Identificación por Radio Frecuencia (RFID) se dio principalmente para poder transmitir la identificación de un objeto (similar a un número de serie), esta tecnología se encuentra dentro de un grupo llamado Auto ID (automatic identification o identificación automática) es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector o reader vinculado a un equipo de cómputo, se comunica a través de una antena con un transponder (también conocido como tag o etiqueta) mediante ondas de radio. [7]

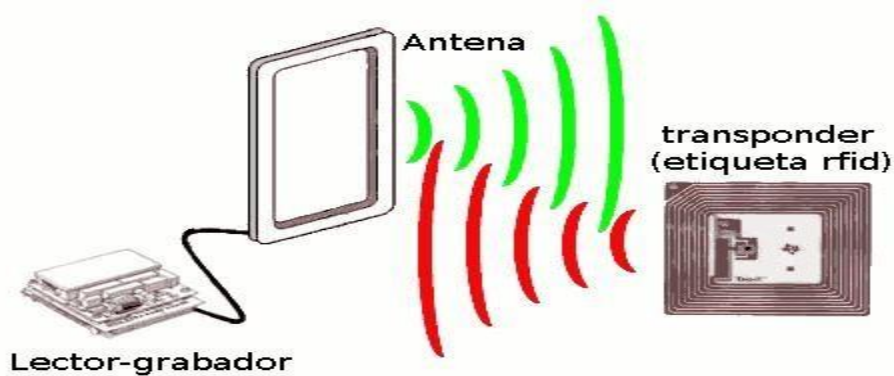


Figura 6. Sistema básico RFID.

Fuente: <https://tsf-info.net/rfid.html>

Las etiquetas RFID (RFID tags en inglés) son dispositivos de un tamaño ligeramente pequeño, similares a una pegatina, que pueden ser pegadas o adheridas a cualquier objeto o producto, un animal o una persona. Contiene antenas para poder realizar la escritura y responder a órdenes por radiofrecuencia desde un emisor-receptor RFID. Las etiquetas pasivas no necesitan alimentación eléctrica interna, mientras que las activas si lo requieren. [9]

RFID Tags están disponibles en diversos formatos y tamaños. Existen en el formato de pastillas, argollas, tarjetas, rectangulares y otros, y los materiales utilizados para el encapsulamiento, además pueden ser de plástico o vidrio. El tipo de RFID tag es definido conforme a la aplicación, ambiente de uso y performance. [9]

La principal característica de los tags pasivos es la ausencia de baterías. Utilizan la energía del enlace de RF para extraer su potencia de activación mediante el uso de circuitos harvesters o rectificadores. La energía recolectada es almacenada en bancos de capacitores integrados al chip. La comunicación entre el tag y lector se lleva a cabo mediante modulación backscatter. La capacidad de memoria es limitada debido a la restricción de potencia. Poseen dimensiones muy reducidas y son planos, con un área aproximada de 15cm x 5 cm. [14]

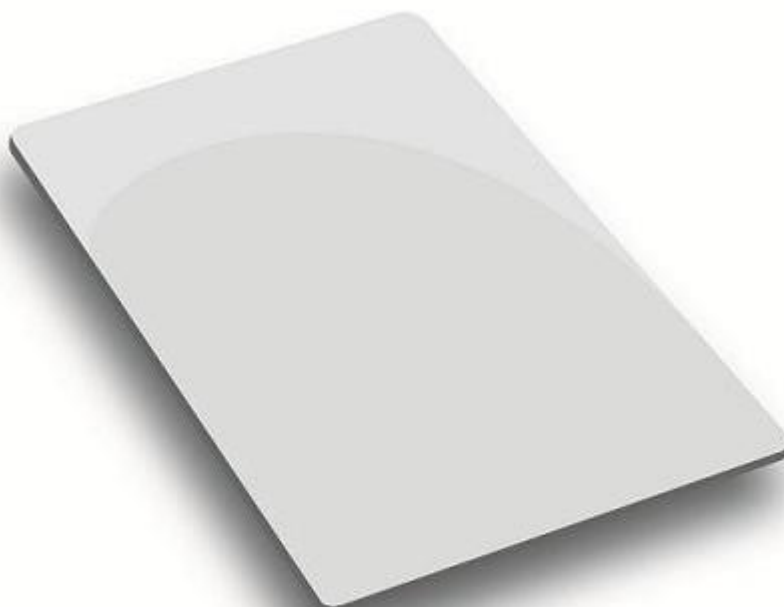


Figura 7. Tag RFID Pasivo modelo tarjeta.

Fuente : <https://www.toshibatec-tsis.com/etiquetas-rfid.html>

Los tags activos utilizan baterías para su funcionamiento. Por lo tanto, su rango de lectura, capacidad de memoria y funciones que realiza se incrementan considerablemente en comparación con su contraparte pasiva. Sus dimensiones son mayores.



Figura 8. Tag RFID Activo modelo SYTAG245-2S.

Fuente:https://www.kimaldi.com/productos/sistemas_rfid/lector_de_tarjetas_rfid_y_tags_activos/tags_rfid_activos/tag_rfid_activo_sytag245_2s/

Esta tecnología que existe desde los años 40, se ha utilizado y se sigue utilizando para múltiples aplicaciones incluyendo casetas de peaje, control de acceso, identificación de ganado y tarjetas electrónicas de transporte.

En los últimos años, la tecnología RFID ha entrado al "mainstream" tecnológico gracias a su creciente difusión en aplicaciones de cadena de suministro motivada por las iniciativas de las cadenas de autoservicio y departamentales. [3]

5.3.1 Elementos de un sistema RFID

Un sistema RFID contiene los siguientes elementos: transponder o tags y lector/escritor.

- **TRANSPONDER:** Es un componente por lo general pasivo y sin batería, compuesto por un circuito integrado (chip) y una antena. El lector, también compuesto por una antena, emite un campo electromagnético. Cuando el transponder entra en el campo de acción de lector, absorbe energía electromagnética del propio lector, que se transforma en energía eléctrica que

carga un condensador cuya energía se utiliza para transmitir al mismo lector su código de identificación. Las transponders constan de una memoria interna que varía, según el tipo de modelo, de unas decenas a unos miles de bytes. Los mismos pueden ser de los siguientes tipos:

Sólo lectura: el código con el que son cargadas es único y se lo personaliza durante la producción.

Lectura y escritura: la información que se guarda en el transponder puede modificarla el lector. Existen los transponders de tipo pasivo y los transponders de tipo activo, con una batería incorporada de por vida que permiten una recepción y transmisión a distancias mayores, al amplificar la señal saliente del transponder aumentando su potencia.

- **LECTOR/ESCRITOR:** Se compone de un circuito que emite energía electromagnética a través de una antena, y una electrónica que recibe y decodifica la información enviada por el transponder y la envía al sistema de captura de datos. De ser el lector fijo, la conexión se produce en serie RS-232, si el lector es portátil la conexión al sistema puede ser del tipo lote. En aplicaciones como el control de acceso, algunos lectores están dotados de una memoria que puede actualizarse desde cualquier computadora que se encuentre conectada a los lectores. [4]



Figura 9. Elementos de un sistema RFID.
Fuente: <https://www.ears.com.ar/tecnologia-rfid/>

5.3.2 Principios básicos de funcionamiento de un sistema RFID.

En la tecnología RFID los principios de operación del circuito consta de 2 elementos principales: la etiqueta y los lectores.

La etiqueta es el elemento que está unido al objeto o producto al que se va a identificar. El lector o también llamado Reader es el mecanismo que permite leer a la etiqueta pegada en el objeto a identificar, logrando así un intercambio de información contenida en la etiqueta, de acuerdo a la programación de la misma.

Cuando el RFID Tag pasa entre la zona electromagnética generada por la antena, es detectado por el lector. El lector decodifica los datos que están codificados en el RFID Tag, pasándolos por la computadora para realizar su procesamiento. [10]

5.3.3 ¿Cómo funcionan los RFID Tags?

La etiqueta RFID, que contiene los datos de identificación del objeto al que se encuentra adherido, genera una señal de radiofrecuencia con dichos datos. Esta señal puede ser captada por un lector RFID, el cual se encarga de leer la información y pasársela, en formato digital, a la aplicación específica que utiliza RFID. [11]

5.3.4 Evolución de los sistemas RFID.

La tecnología RFID ha evolucionado la identificación a distancia. Han pasado varios años desde el nacimiento de la tecnología, pero en estos últimos años con la aparición de grandes organizaciones han hecho que se produzca un gran número de estos dispositivos para el control de objetos ya que más se enfocan en la tecnología RFID en los mercados.

El futuro del sistema RFID parece ser esperanzador, ya que en un mundo basado en el poder de la información esta tecnología provoca una gran ventaja para el almacenamiento de la misma. El interés por el comercio virtual parece que tiene su principal protector en estos sistemas en la gestión de todo el proceso. Pero para estas nuevas aplicaciones se necesita un gran desarrollo de la tecnología.

La evolución de los sistemas de identificación de localización basados en la tecnología RFID se presenta de la siguiente forma.

- 1940 - 1950: En esta etapa de evolución se presenta la creación de radar y lo usaban para el control de los movimientos de los aviones en la Segunda Guerra Mundial.
- 1950 - 1960: Primeras averiguaciones a la tecnología de RFID el funcionamiento y poniendo a primeros experimentos en laboratorios.
- 1960 - 1970: Desarrollo de la teoría de RFID. Se comienzan a desarrollar aplicaciones a diversas áreas de trabajo.
- 1970 - 1980: Un gran disparo en el desarrollo de sistemas de RFID. Se aumenta la velocidad en el desarrollo de las pruebas de la tecnología. Aplicando los primeros usos en empresas industriales innovadoras de sistemas de RFID.
- 1980 - 1990: Nacimiento de las primeras aplicaciones en el proceso comercial tecnológico.

- 1990 - 2000: Surgimiento de estándares internacionales. Amplio despliegue de tecnologías de RFID. El sistema RFID pasa a formar parte de la vida cotidiana.
- 2001 Invierte en estudios y desarrollos en RFID y desarrolla lectores de RFID de baja frecuencia operando con equipos para leer chips en llantas de Tráiler. [11]

5.3.5 Módulo RFID RC 522

Este módulo utiliza un sistema de modulación y demodulación de 13.56MHz, frecuencia que en la actualidad utiliza la tecnología RFID.

EL módulo se comunica por SPI, por lo que se puede implementar con cualquier microcontrolador con interfaz SPI, como ejemplo el Arduino Uno. [6]

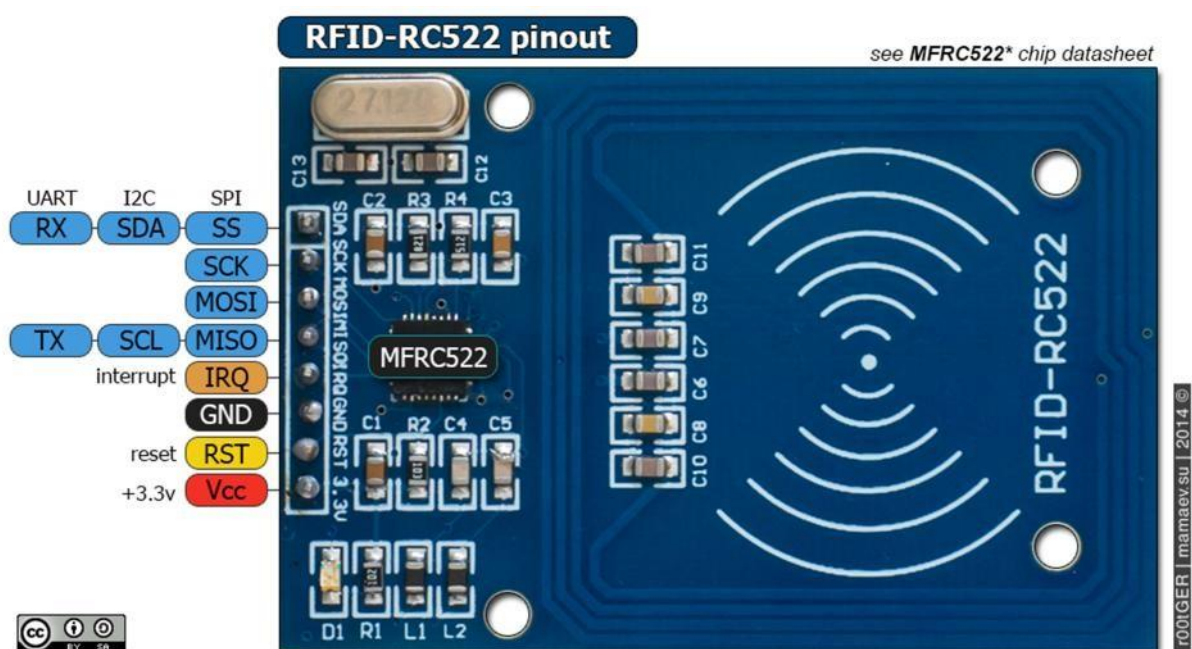


Figura 10. Módulo RC-522 para Arduino.

Fuente: <https://saber.patagoniatec.com/2016/07/lector-de-tarjetastags-rfid-rc522-13-56mhz-nfc/>

5.3.5.1 Comunicación SPI

SPI (Serial Peripheral Interface) es básicamente un bus de comunicación a nivel de circuitos integrados. La transmisión de datos se realiza en serie, es decir un bit después de otro. El bus SPI se define mediante 4 pines:

SCLK o SCK: Señal de reloj del bus. Esta señal rige la velocidad a la que se transmite cada bit.

MISO (Master Input Slave Output): Es la señal de entrada a nuestro dispositivo, por aquí se reciben los datos desde el otro integrado.

MOSI (Master Output Slave Input): Transmisión de datos hacia el otro integrado.

SS o CS: Chip Select o Slave Select, habilita el integrado hacia el que se envían los datos. Esta señal es opcional y en algunos casos no se usa. [16]

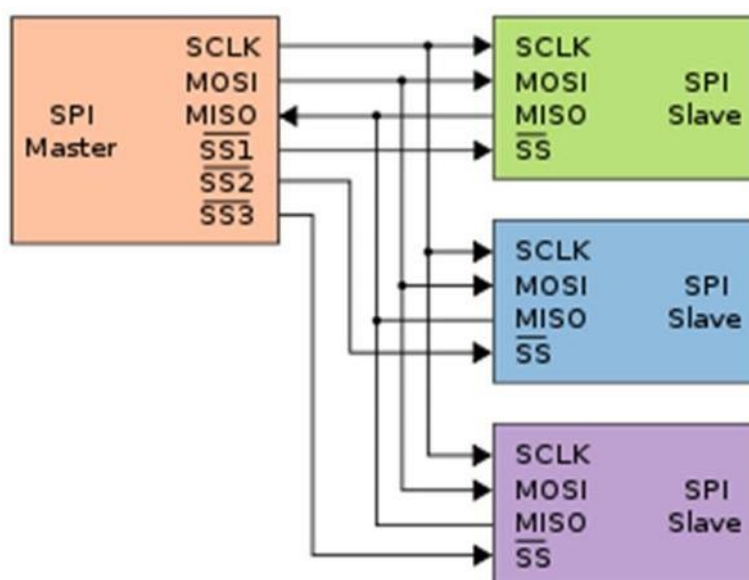


Figura 11. Conexión SPI.

Fuente: https://es.wikipedia.org/wiki/Serial_Peripheral_Interface

ARDUINO UNO	RFID RC552
DIGITAL PIN #10	SDA
DIGITAL PIN #13	SCK
DIGITAL PIN #11	MOSI
DIGITAL PIN #12	MISO
N/A	IRQ
POWER GND	GND
DIGITAL PIN #5	RST
POWER 3.3 V	3.3 V

Figura 12. Tabla de pines del Módulo RC-522 a Arduino.

Fuente: <https://saber.patagoniatec.com/2016/07/lector-de-tarjetastags-rfid-rc522-13-56mhz-nfc/>

5.4 Bases de Datos

Una Base de Datos o Banco de Datos es un conjunto de datos pertenecientes a un mismo contexto almacenados en un orden específicamente descrito; o puede considerarse una biblioteca ya que ambas almacenan información, estas tienen un orden y siempre que se necesite algún tipo de información la vamos a poder encontrar de una manera más fácil y eficiente. Estas Bases de Datos fueron creadas debido al gran crecimiento tecnológico de la informática y la electrónica, antes las Bases de Datos se almacenaban en papel, esto no era un sistema muy eficiente ya que se podía perder información y era complicado de encontrar cuando se necesitaba algún tipo de información en específico, debido a esto se crearon Bases de Datos en formato digital para evitar el desperdicio de papel y facilitar la búsqueda de información. [5]

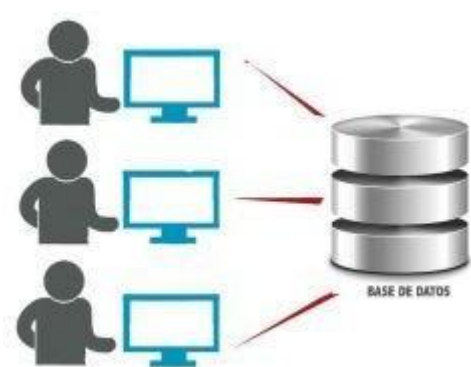


Figura 13. Funcionamiento de una Base de Datos.

Fuente: <https://www.significados.com/base-de-datos/>

5.4.1 Herramientas para crear Bases de Datos

Los programas o herramientas para crear Bases de Datos organizan y almacenan los datos de tal manera que puedan crear y buscar cualquier tipo de información de manera simple y sencilla para ello contamos con los siguientes programas más conocidos:

- **Access:**

Access es un sistema de Base de Datos personal de Microsoft. Se trata de un producto de software orientado hacia lo visual, lo que hace que quienes no sean programadores puedan crear Bases de Datos útiles con facilidad.

- **Visual FoxPro:**

FoxPro es un sistema de Base de Datos relacional, también producido por Microsoft, que está estrechamente unido a su lenguaje de programación. FoxPro es menos amigable que las Bases de Datos de usuario final, lo que requiere más conocimientos técnicos que Access.

- **MySQL Database:**

MySQL es una Base de Datos basada en un servidor que permite a varios usuarios acceder a múltiples Bases de Datos. El software funciona en múltiples plataformas, incluyendo la mayoría de las variedades de UNIX y Windows. Ofrece usabilidad de primer plano limitada y está diseñado como un servidor de Base de Datos back-end.

- **Oracle:**

La Oracle es otra Base de Datos escalable a nivel empresarial. La Base de Datos de SQL soporta Bases de Datos corporativos distribuidos, que permiten al usuario acceder a los datos de forma local o desde Bases de Datos remotas en una transacción transparente. Las bases de datos distribuidas ayudan a superar las limitaciones físicas de un entorno informático físico. [12]

5.5 jQuery



Figura 14. JQuery muestra que es fácil de programar con JavaScript.

Fuente:[https://www.spleasons.com/lesson/jquerytutorials/#SPLessonPhoto\[myGallery\]/1/](https://www.spleasons.com/lesson/jquerytutorials/#SPLessonPhoto[myGallery]/1/)

jQuery es considerado un Framework de JavaScript, o ambiente de desarrollo, lo que no es más que un conjunto de utilidades las cuales no necesitan ser programadas, de hecho, ya fueron programadas, probadas y podemos utilizarlas de una manera muy sencilla, jQuery nos permite agregar efectos y funcionalidades complejas a nuestro sitio web, como, por ejemplo: galerías de fotos dinámicas y elegantes, validación de formularios, calendarios, hacer aparecer y desaparecer

elementos en nuestra página y muchas otras poderosas opciones. Una ventaja sin duda es la posibilidad que nos brinda de trabajar con AJAX (envío de información específica), sin preocuparnos de los detalles complejos de la programación.

Framework se podría traducir al español como “Marco de desarrollo”, En nuestro caso el marco sería en JavaScript, jQuery es un marco de desarrollo el cual contiene funcionalidades y librerías pre-desarrolladas.

Ajax es un conjunto de métodos y técnicas que permiten intercambiar datos con un servidor y actualizar partes de páginas web sin necesidad de recargar la página completamente. [13]

5.6 EXCEL y su manejo de datos

Microsoft Excel es un software de aplicación publicado por la empresa Microsoft, que brinda soporte digital a las labores contables, financieras, organizativas y de programación, mediante hojas de cálculo. [16]

El manejo de datos con EXCEL es mucho más fácil ya que es una Base de Datos que adicional a esta se puede manipular de la manera que se desee ya que este cuenta con herramientas apropiadas para el manejo de datos, como son fórmulas, funciones y Visual Basic, que nos permite ordenar los datos a los requerimientos del programa. [17]

6. Desarrollo del Proyecto de Titulación

Primero se establecen los requerimientos que se necesitan en el Sistema de Control de Registro de Asistencia, las tecnologías con las que se dispone en el mercado local, así como qué tipo de herramientas son las más apropiadas para realizar este sistema, después se procede a diseñar el sistema con las herramientas que son las más apropiadas para el sistema, luego se crea la Base de Datos que contendrá la información de registro de asistencia de cada laboratorio en la cual se mostraran los datos en una interfaz gráfica amigable con el usuario, entonces se realiza la implementación del sistema en los laboratorios específicos de la Institución. Por último, se procede a realizar las pruebas necesarias de funcionamiento dando por terminado con éxito el proyecto cumpliendo con los objetivos del mismo.

6.1 Requerimientos del sistema y tecnologías disponibles en el mercado local

Para realizar el sistema se necesita la representación de comportamiento el cual empieza con una Base de Datos que registrará el número de identificación de la tarjeta (Tag RFID dispositivo que contiene una antena para lectura de por radiofrecuencia), la fecha y hora en la cual el alumno pasa la tarjeta por el lector, además la base de datos contendrá la información de cada estudiante la cual constará de sus nombres y apellidos, la Carrera que estudia y su número de identificación, finalmente para poder visualizar el registro de asistencia se creará una página HTML la cual leerá la información de la Base de Datos y nos la mostrará en una tabla y para poder llevar la información se la podrá descargar en formato XML y PDF para su fácil revisión, el cual utilizará cualquier PC que este dentro de la red LAN Administrativa del Instituto Tecnológico Superior Sudamericano Quito, para poder acceder mediante un usuario y contraseña a la información registrada en el servidor que contiene la Base de Datos.

En el mercado local existen varias opciones de microcontroladores, el más conveniente para realizar este sistema es Arduino el cual cuenta con diferentes versiones hay desde el Arduino nano, Uno, mini, mega, etc.

Pero el más eficiente para realizar este sistema es el Arduino UNO el cual cumple con las necesidades del sistema, para realizar la lectura de las tarjetas se utilizará un lector compatible con Arduino UNO; existen diferentes lectores desde el RC-522, PN532, módulo lector RFID 125 kHz el más adecuado para este sistema es el RC-522. Para poder enviar la información a la Base de Datos se utilizará una shield ethernet creada especialmente para Arduino, se utilizará la Base de Datos llamada MySQL por la facilidad de enviar la información desde Arduino, para la

visualización de la información guardada en la base se utilizará una página HTML con una librería llamada JQuery la cual ordena la información y cuenta con un buscador para poder realizar una búsqueda específica.

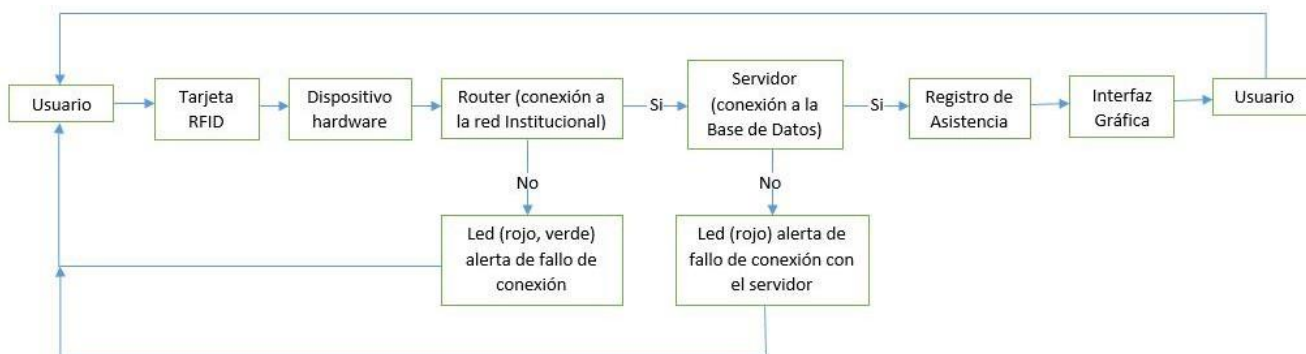


Figura 15. Diagrama de funcionamiento del sistema de registro de asistencia.

Fuente: el autor.

6.2 Diseño del sistema

Una vez definido los requerimientos del sistema y sus tecnologías disponibles, se procede a armar el dispositivo de lectura y envío de información a la Base de Datos, cuando el dispositivo envíe la información correctamente se procederá a crear una base de datos con la información de cada estudiante y finalmente se la podrá visualizar en una página HTML, la cual contará con una herramienta de búsqueda e impresión de los datos en formato EXCEL y PDF.

Para la creación del sistema necesitamos diseñar un esquema de su funcionamiento que nos muestre como el sistema tiene que comportarse.



Figura 16. Esquema de diseño del sistema.

Fuente: el autor.

6.3 Implementación del sistema

Para el envío de información a la Base de Datos se necesita construir un dispositivo hardware el cual nos permita realizar la lectura de las tarjetas y enviar la información dentro de los mismos a la Base de Datos, para realizar este proceso se procede a montar la Ethernet Shield sobre el Arduino Uno, se debe conectar los pines del Módulo RC-522 a las entradas de la Ethernet Shield como se muestra a continuación en la figura 17, para la conexión de los pines del Módulo RC-522 se puede consultar la figura 12 o el Anexo 3 el cual detalla la información de conexión.

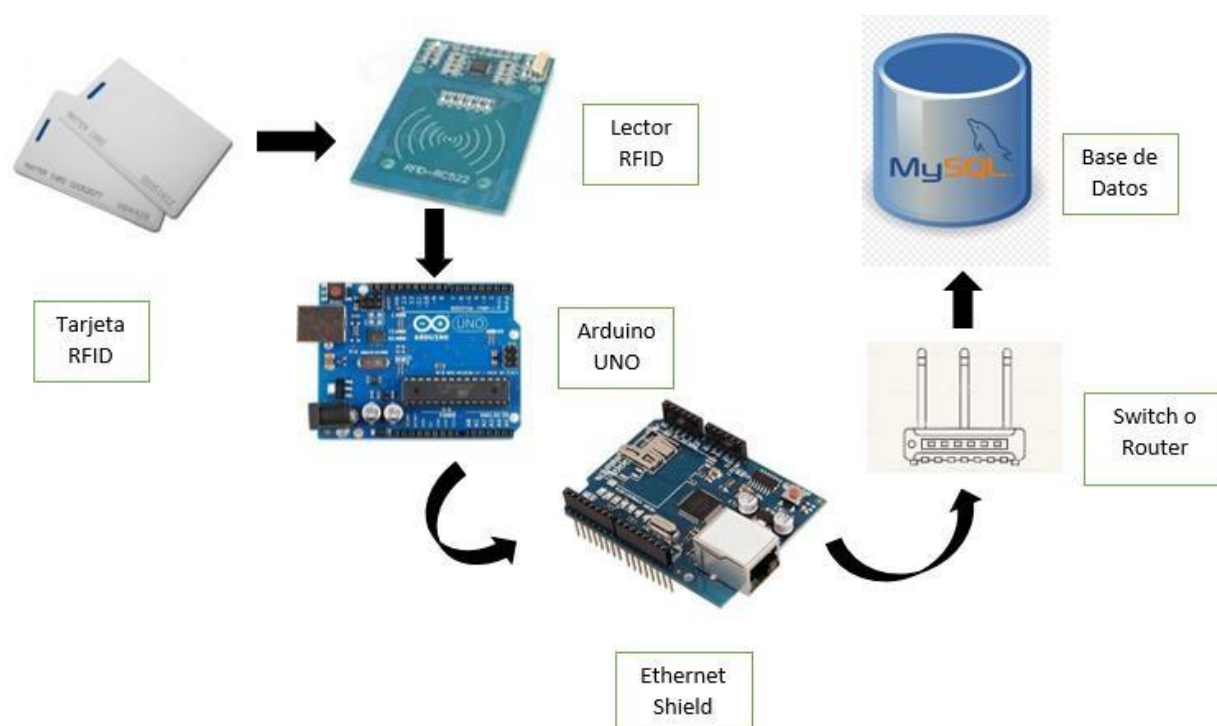


Figura 17. Dispositivo hardware de envío de información a la Base de Datos

Fuente: el autor.

El IDE de Arduino es un entorno de código abierto, Arduino hace posible escribir el código y cargarlo a la placa de Arduino. Esta plataforma está habilitada para Windows, Mac OS X, Linux y Android. El entorno está escrito en Java y basado en Processing, avr-gcc y otros programas también de código abierto. El entorno contiene un editor de textos para escribir el código, en donde indica errores gramaticales de programación, una consola para interactuar con el hardware de Arduino, una barra de herramientas y también algunos menús. Este IDE hace posible la carga de programas a la placa Arduino e interactuar con el mismo.

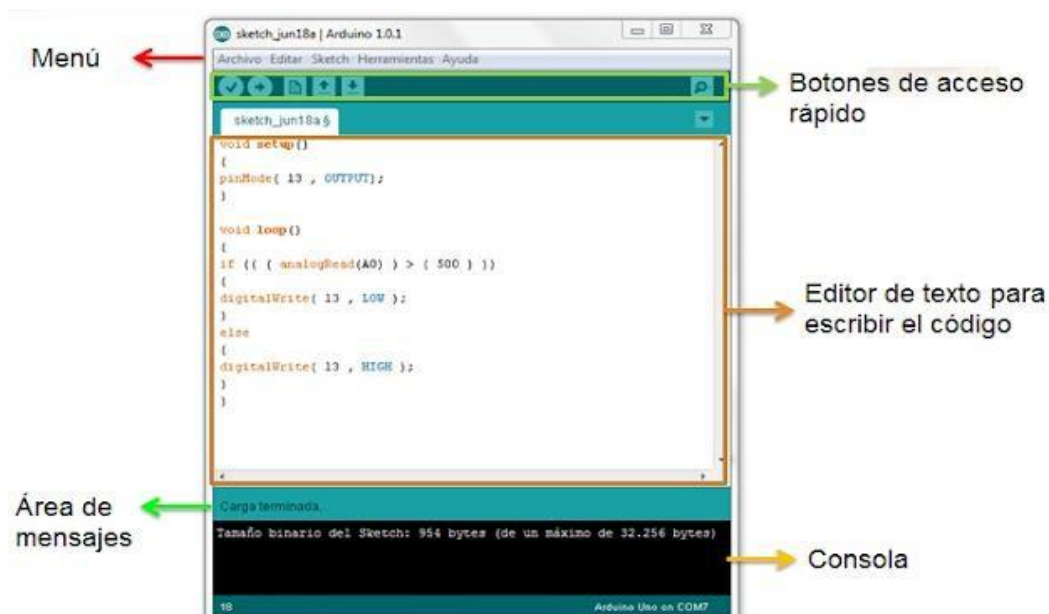


Figura 18. IDE para programar e interactuar con el hardware de Arduino.

Fuente: <http://solorobotica.blogspot.com/2012/07/programacion-de-arduino-elide-de.html>

Para que el dispositivo hardware pueda enviar la información es necesario programarlo en el IDE de Arduino y configurarlo para que pueda realizar la lectura y envío de información en el momento que se pase la tarjeta por el dispositivo hardware.

El sketch de configuración tiene primero la inicialización de librerías para la utilización de la Ethernet Shield y el Módulo RC-522, configuración de los pines para la utilización de leds, zumbador, pines de la Ethernet Shield y RC-522.

```
#include<SPI.h>
#include<MFRC522.h>
#include<SoftwareSerial.h>
#include <Ethernet.h>
#define SS_PIN 4 //FOR RFID SS PIN BECASUSE WE ARE USING BOTH ETHERNET SHIELD AND RS-522
#define RST_PIN 9

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xBB};
byte server[] = {192 , 168 , 1, 3};
byte ip[] = {192, 168, 33, 239};
byte DNS[] = {192, 168, 33, 1};
byte gateway[] = {192, 168, 33, 1};
byte subnet[] = {255, 255, 255, 0};

EthernetClient client;
MFRC522 rfid(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
byte i;

int led_verde = 5;
int zumbador = 6;
int led_rojo = 7;
```

Figura 19. Configuración de pines y librerías necesarias.

Fuente: el autor.

Se debe configurar los pines como salida (OUTPUT) para poder indicar cuando se debe encender o apagar de los leds y el zumbador, se abre el puerto serial con 9600 para poder enviar mensajes a la PC, se inicializa la comunicación SPI para poder configurar la Ethernet Shield y el módulo RC-522, se continua configurando la Ethernet Shield para la conexión a la red administrativa del Instituto Tecnológico Superior Sudamericano Quito, si la configuración falla los leds indicadores se mantendrán encendidos al mismo tiempo, seguido de un mensaje de error que se lo puede ver cuando se conecte la placa Arduino a la PC y se habrá el monitor serie que es donde muestra los mensajes que envía la placa Arduino, si la conexión es correcta se enciende el led verde y muestra un mensaje indicando que la conexión se estableció correctamente a la red.

```

void setup()
{
  pinMode(zumbador , OUTPUT);
  pinMode(led_verde , OUTPUT);
  pinMode(led_rojo , OUTPUT);
  Serial.begin(9600);

  SPI.begin();
  rfid.PCD_Init();
  for (byte i = 0; i < 6; i++)
  {
    key.keyByte[i] = 0xFF;
  }
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    digitalWrite(led_verde , HIGH);
    digitalWrite(led_rojo , HIGH);
    Ethernet.begin(mac, ip, DNS, gateway, subnet);
  }
  delay(1000);
  Serial.println("connecting...");
  digitalWrite(led_verde , HIGH);
}

```

Figura 20. Configuración de Ethernet Shield y comprobación de conexión a la red Administrativa.

Fuente: el autor.

Se continua con la lectura de las tarjetas cuando estas se acerquen a una distancia de 3 cm o menos del Módulo RC-522 este procederá a la lectura de la identificación de la tarjeta y la guarda en una variable , continua deteniendo la lectura hasta que se pase otra tarjeta cerca del módulo RC-522, después de guardar el número de identificación de la tarjeta continua hacia la función *Void Sending_To_BD* donde comprueba si hay conexión con el servidor, de no haberla hay un pitido de 1.5 segundos y se enciende el led de color rojo , si hay conexión con el servidor se envía un mensaje hacia el monitor mostrando conexión, se indica la dirección donde se encuentra el archivo que enviará la información hacia la Base de Datos, se guarda la identificación de la tarjeta en una variable llamada *tarjeta*, seguido de un pitido y se mantiene encendido el led de color verde.


```

void loop()
{

  if (!rfid.PICC_IsNewCardPresent())
    return;
  if (!rfid.PICC_ReadCardSerial())
    return;

  Serial.print("your card no :");
  for (int s = 0; s < 4; s++)
  {
    Serial.print(rfid.uid.uidByte[s]);
    Serial.print(" ");
  }
  Serial.println();
  Sending_To_DB();

  rfid.PICC_HaltA(); rfid.PCD_StopCryptol(); return;

  // Halt PICC
  rfid.PICC_HaltA();
  // Stop encryption on PCD
  rfid.PCD_StopCryptol();
}

```

Figura 21. Lectura de tarjetas.

Fuente: el autor.

```

void Sending_To_DB() //CONNECTING WITH MYSQL
{
  if (client.connect(server, 8080)) {
    Serial.println("connected");
    // Make a HTTP request:
    client.print("GET /arduino_401/data404.php?"); //YOUR URL
    client.print("tarjeta=");
    for (int i = 0; i < 4; i++)
    {
      client.print(rfid.uid.uidByte[i]);
    }
    digitalWrite(zumbador , HIGH);
    delay(50);
    digitalWrite(zumbador , LOW);
    digitalWrite(led_verde , HIGH);
    digitalWrite(led_rojo , LOW);
    client.println();
    client.stop();
  }
  else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed");
    digitalWrite(zumbador , HIGH);
    delay(1500);
    digitalWrite(zumbador , LOW);
    digitalWrite(led_verde , LOW);
    digitalWrite(led_rojo , HIGH);
  }
}
}

```

Figura 22. Conexión a la Base de Datos y envío de identificación de tarjeta.

Fuente: el autor.

Para que la Base pueda recibir los datos enviados por el dispositivo hardware es necesario crear un enlace que permita la conexión con el servidor y al final este guarde los datos en la Base.

Primero se crea la conexión del servidor con la Base de Datos, se continua con la creación del archivo que guardará los datos dentro de la Base.

El archivo `connection.php` contiene un usuario y contraseña de phpMyAdmin el cual es el gestor de las Bases de Datos, también se hace referencia a `localhost` porque es el host o pc donde se está ejecutando el programa en este caso está dentro del mismo servidor, por último, se referencia el nombre de la Base de Datos a la cual se va enlazar y se realiza una consulta SQL al servidor.

```
connection.php
1 <?php
2 $username = "arduino";
3 $pass = "1234";
4 $host = "localhost";
5 $db_name = "registro_asistencia_alumnos";
6 $con = mysqli_connect ($host, $username, $pass);
7 $db = mysqli_select_db ( $con, $db_name );
8 ?>
```

Figura 23. Archivo de conexión con la Base de datos.

Fuente: el autor.

Se debe continuar creando un archivo para guardar los datos dentro de la Base, primero se incluye el archivo de `conecction.php`, después se debe crear una sentencia SQL para guardar los datos de la variable `tarjeta` del sketch de Arduino en el campo también llamado `tarjeta` de la Base `registro_asistencia_alumnos`.

```
data404.php
1 <?php
2 include ('connection.php');
3 $sql_insert = "INSERT INTO data4 (tarjeta) VALUES ('".$_GET["tarjeta"]."')";
4 if(mysqli_query($con,$sql_insert))
5 {
6 echo "Done";
7 mysqli_close($con);
8 }
9 else
10 {
11 echo "error is ".mysqli_error($con );
12 }
13 ?>
```

Figura 24. Archivo para guardar la identificación de la tarjeta en la Base de Datos.

Fuente: el autor.

6.3.1 Creación de la Base de Datos y su interfaz gráfica

Para la creación de la Base de Datos se utilizará el paquete de software libre llamado xampp el cual contiene un sistema de gestión de Base de Datos MySQL. Para la utilización de este se necesita de una computadora la cual cuente con Windows o Linux.



Figura 25. Funcionamiento de la Base de Datos con la interfaz gráfica.

Fuente: el autor.

Primero se creó la Base de Datos llamada *registro_asistencia_alumnos* con la herramienta MySQL Database.

Bases de datos

Crear base de datos

registro_asistencia_alumnos latin1_swedish_ci

Filtros

Que contengan la palabra:

Base de datos	Cotejamiento	Acción
<input type="checkbox"/> datatables	latin1_swedish_ci	Seleccionar privilegios
<input type="checkbox"/> ethernet1	latin1_swedish_ci	Seleccionar privilegios
<input type="checkbox"/> information_schema	utf8_general_ci	Seleccionar privilegios
<input type="checkbox"/> instituto sudmaericano	latin1_swedish_ci	Seleccionar privilegios

Figura 26. Creación de la Base llamada registro asistencia alumnos.

Fuente: el autor.

Se continúa creando la tabla *data* con los campos *fecha* este será de tipo `TIMESTAMP` para que al momento de que se registre una asistencia se guardará la fecha y hora actual del servidor, campo *id* será tipo `INT` porque es un número de auto incremento, el campo *tarjeta* será tipo `VARCHAR` para poder registrar el número de identificación de la tarjeta, estos son los campos que se va a necesitar al momento de que el dispositivo hardware envíe la información hacia esta tabla, específicamente hacia el campo *tarjeta* donde se almacenará nuestra identificación de la tarjeta dentro de la Base de Datos.

Nombre de la tabla: Agregar columna(s)

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice
fecha	TIMESTAMP		Ninguno			<input type="checkbox"/>	---
id	INT		Ninguno			<input type="checkbox"/>	---
tarjeta	VARCHAR		Ninguno			<input type="checkbox"/>	---

Figura 27. Creación de la tabla llamada data con sus campos necesarios.

Fuente: el autor.

Se crea la tabla *estudiantes* con los campos *carrera*, *id_tarjeta* y *nombre*. Se crea esta tabla para aquí poder almacenar los datos de los estudiantes los cuales en el campo *carrera* contendrá la Carrera que el estudiante esté estudiando, el campo *id_tarjeta* contendrá su número de identificación en la tarjeta, el campo *nombre* contendrá sus dos apellidos y nombres.

Figura 28. Creación de la tabla llamada estudiantes con sus campos necesarios.

Fuente: el autor.

En esta captura de pantalla se muestra como estará compuesta la tabla *data* y como estarán almacenados sus datos todo esto se realizará en la Base de Datos.

Opciones	id	fecha	tarjeta
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	332	2019-01-27 22:00:38	418621443
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	333	2019-01-27 22:00:41	410621443
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	334	2019-01-27 22:00:43	418621443
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	335	2019-01-27 22:00:45	418621443
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	336	2019-01-27 22:00:46	418621443
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	337	2019-01-27 22:00:58	10289101172

Figura 29. Pantalla de ejemplo de información de la tabla data.

Fuente: el autor.

En esta captura de pantalla se muestra como estará compuesta la tabla *estudiantes* y como estará ingresada la información, su número de identificación en la tarjeta, sus apellidos y nombres, y la Carrera que estudia.

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0000 segundos.)

```
SELECT * FROM `estudiantes`
```

id_tarjeta	nombre	carrera
10289101172	junior rojas	sistemas
123456	juan	ambiente
418621443	karen rojas dayana	sistemas
1234567890	kevin andrango	sistemas

Figura 30. Pantalla de ejemplo información de la tabla estudiantes.

Fuente: el autor.

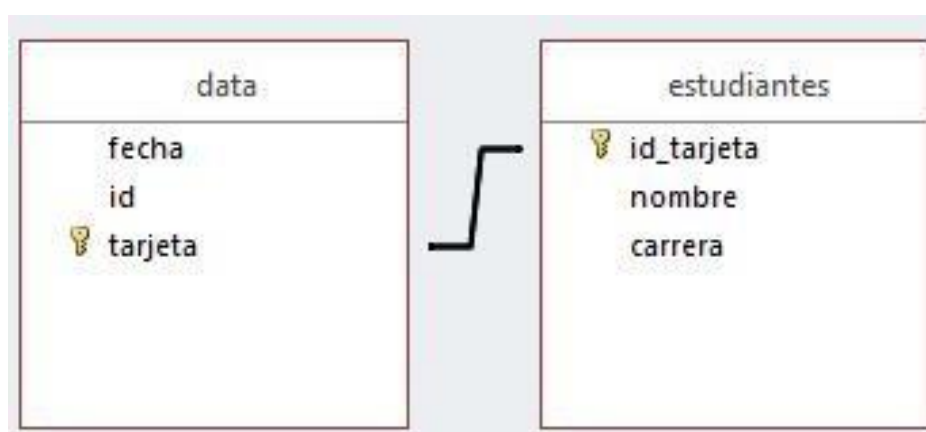


Figura 31. Relación entre tablas.

Fuente: el autor.

Para la creación de la interfaz gráfica se utiliza una página HTML y lenguaje de programación Java Script.

Para la visualización de los estudiantes que asisten a clases se utilizará la Librería llamada jQuery que permite mostrar la información de una Base de Datos en una tabla, esto lo realiza mediante una consulta SQL, para que no haya problemas de visualización todo esto lo realizamos dentro de una página HTML, la ventaja de realizarla en esta página es que se puede manipular mediante programación para poder crearla visualmente agradable a la vista del usuario.

Primero se agrega un encabezado con el nombre del aula, se agrega las librerías necesarias para el datatable que es donde van a estar almacenados los datos de la Base.

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <head>
6 <title ">Asistencia de esudiantes Aula 301</title>
7 <link rel="stylesheet" href="css/bootstrap.min.css">
8 <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">
9 <link rel="stylesheet" href="css/estilos.css">
10 <!-- Buttons DataTables -->
11 <link rel="stylesheet" href="css/buttons.bootstrap.min.css">
12 <link rel="stylesheet" href="css/font-awesome.min.css">
13 </head>
14 <body>
15 <body style="background-color: black;">
16 <div class="row fondo">
17 <div class="col-sm-12 col-md-12 col-lg-12">
18 <h1 style="color :black;"class="text-center text-uppercase">Asistencia del Aula 301</h1>
19 </div>
20 </div>
21

```

Figura 32. Agregado de librerías para datatable y título de la página HTML.

Fuente: el autor.

se continúa agregando el nombre y número de Aula, se crea la datatable llamada *dt_cliente* con los nombres de los encabezados de cada columna los cuales son *Fecha*, *Hora*, *Día*, *Id Tarjeta*, *Nombre*, *Carrera*.

```

22 <div class="row">
23 <div id="cuadro1" class="col-sm-12 col-md-12 col-lg-12">
24 <div class="col-sm-offset-2 col-sm-8">
25 <h3 class="text-center"><small class="mensaje"></small></h3>
26 </div>
27 <div class="table-responsive col-sm-12">
28 <table id="dt_cliente" class="table table-bordered table-hover" cellspacing="0" width="
100%" style="background-color:#006060;" border="1px">
29 <thead>
30 <tr>
31 <th style="color : black ;">Fecha</th>
32 <th style="color : black ;">Hora</th>
33 <th style="color : black ;">Dia</th>
34 <th style="color : black ;">Id Tarjeta</th>
35 <th style="color : black ;">Nombre</th>
36 <th style="color : black ;">Carrera</th>
37 </tr>
38 </thead>
39 </table>
40 </div>
41 </div>
42 </div>
43 <div>
44 </div>

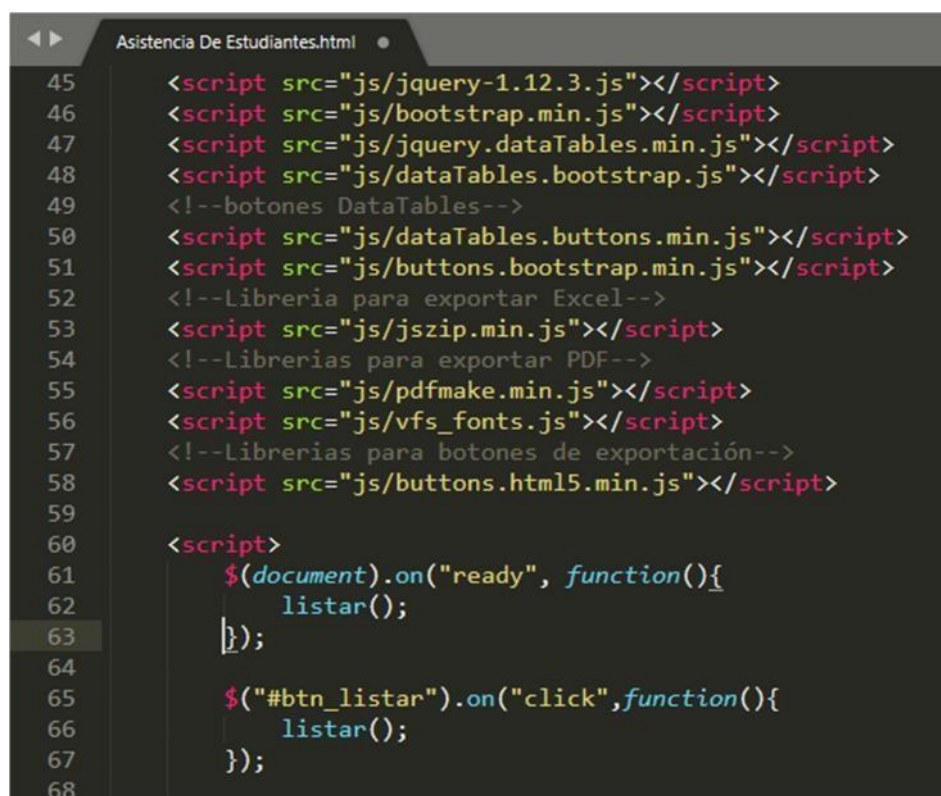
```

Figura 33. Creación de la tabla *dt_cliente* y sus columnas con sus respectivos nombres.

Fuente: el autor.

Se continúa agregando las librerías de lenguaje JavaScript donde están los botones para exportar a Excel y PDF, se sigue con la función de lectura de los datos se crea la función *listar* () esto para poder realizar la publicación de los datos capturados por el archivo *listar.php* estos

datos son capturados mediante AJAX (captura de datos de un servidor) y son puestos en la tabla mediante el método POST (envío de datos de forma invisible para el cliente).



```

45 <script src="js/jquery-1.12.3.js"></script>
46 <script src="js/bootstrap.min.js"></script>
47 <script src="js/jquery.dataTables.min.js"></script>
48 <script src="js/dataTables.bootstrap.js"></script>
49 <!--botones DataTables-->
50 <script src="js/dataTables.buttons.min.js"></script>
51 <script src="js/buttons.bootstrap.min.js"></script>
52 <!--Libreria para exportar Excel-->
53 <script src="js/jzip.min.js"></script>
54 <!--Librerias para exportar PDF-->
55 <script src="js/pdfmake.min.js"></script>
56 <script src="js/vfs_fonts.js"></script>
57 <!--Librerias para botones de exportación-->
58 <script src="js/buttons.html5.min.js"></script>
59
60 <script>
61     $(document).on("ready", function(){
62         listar();
63     });
64
65     $("#btn_listar").on("click",function(){
66         listar();
67     });
68

```

Figura 34. Agregado de librerías para exportar datos a Excel y PDF, lectura de los datos.

Fuente: el autor.

Se crea dos funciones para poder deslizar la página hacia arriba y hacia abajo para poder visualizar mejor los datos, se ordena los datos de la primera columna la cual va a ser las fechas, se crea la función AJAX para capturar los datos, se declara las columnas que se va a leer en el archivo listar.php las cuales son *fecha, hora, dia_semana, id_tarjeta, nombre, carrera*, se agrega el idioma español a la tabla ya que por defecto es inglés.

Por último, se crea la función *idioma_espanol*, la cual es una función que ya está programada solo hay que buscar en la página oficial de jQuery y agregarlo al código para que la tabla que por defecto está en idioma inglés lo transforme a español.

Todas las librerías deben estar dentro de la misma carpeta donde se crea la página HTML para obtener un orden se crea dos carpetas: la primera llamada *css* dentro de esta se encontrarán todas las librerías cuyas terminaciones sean *.css*, la segunda llamada *js* dentro de esta se encontrarán todas las librerías cuyas terminaciones sean *.js*.


```

69     var listar = function(){
70         $("#cuadro1").slideDown("slow");
71         $("#cuadro2").slideUp("slow");
72         var table = $("#dt_cliente").DataTable({
73             "destroy":true,
74             "order": [[ 0 , "desc" ],[ 1 , "desc" ]],
75             "fnRowCallback": function( nRow, aData, iDisplayIndex, iDisplayIndexFull ) {
76                 $('td', nRow).css('background-color', '#337ab7cc ');
77                 $('td', nRow).css('color', 'black' );
78             },
79
80             "ajax":{
81                 "method":"POST",
82                 "url":"listar.php"
83             },
84             "columns":[
85                 {"data":'fecha',"width": "10%"},
86                 {"data":'hora',"width": "10%"},
87                 {"data":'dia_semana',"width": "10%"},
88                 {"data":'id_tarjeta',"width": "10%"},
89                 {"data":'nombre',"width": "10%"},
90                 {"data":'carrera',"width": "10%"}
91             ],
92             "language":idioma_espanol,
93             "dom":"<'row'<'form-inline' <'col-sm-offset-5'B>>>"
94                 + "<'row' <'form-inline' <'col-sm-1'f>>>"
95                 + "<rt>"
96                 + "<'row'<'form-inline'"
97                 + " <'col-sm-6 col-md-6 col-lg-6'l>"
98                 + "<'col-sm-6 col-md-6 col-lg-6'p>>>",

```

Figura 35. Creación de la función *listar*.

Fuente: el autor.

Se crea los botones para exportar los datos de la tabla en EXCEL y PDF.

```

99         "buttons":[
100             {
101                 extend: 'excelHtml5',
102                 text: '<i class="fa fa-file-excel-o"></i>',
103                 titleAttr: 'Excel'
104             },
105             {
106                 extend: 'pdfHtml5',
107                 text: '<i class="fa fa-file-pdf-o"></i>',
108                 titleAttr: 'PDF'
109             }
110         ]
111     });
112 }
113

```

Figura 36. Creación de botones para exportar a Excel y PDF.

Fuente: el autor.

Para agregar el idioma español a la tabla se obtiene una función creada previamente por los desarrolladores de jQuery la cual solo hay que descargarla y agregarla al código.

```

114     var idioma_espanol={
115         "sProcessing":      "Procesando...",
116         "sLengthMenu":     "Mostrar _MENU_ registros",
117         "sZeroRecords":    "No se encontraron resultados",
118         "sEmptyTable":     "Ningún dato disponible en esta tabla",
119         "sInfo":           "Mostrando registros del _START_ al _END_ de un total de _TOTAL_ registros",
120         "sInfoEmpty":      "Mostrando registros del 0 al 0 de un total de 0 registros",
121         "sInfoFiltered":   "(filtrado de un total de _MAX_ registros)",
122         "sInfoPostFix":    "",
123         "sSearch":         "Buscar:",
124         "sUrl":             "",
125         "sInfoThousands":  ",",
126         "sLoadingRecords": "Cargando...",
127         "oPaginate": {
128             "sFirst":      "Primero",
129             "sLast":       "Último",
130             "sNext":       "Siguiete",
131             "sPrevious":   "Anterior"
132         },
133         "oAria": {
134             "sSortAscending": ": Activar para ordenar la columna de manera ascendente",
135             "sSortDescending": ": Activar para ordenar la columna de manera descendente"
136         }
137     }
138 }
139 </script>
140 </body>
141 </html>

```

Figura 37. Creación de la función idioma_espanol ().

Fuente: el autor.

Para obtener seguridad se hace una comparación de si un usuario a ingresado se pueda visualizar la página caso contrario esta lo re direcciona a la página de inicio del Sistema y también si han pasado 5 minutos esta le pida que vuelva a ingresar el usuario y contraseña.

```

141 <?php
142 session_start();
143
144 if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true) {
145 } else {
146     echo "Inicia Sesion para acceder a este contenido.<br>";
147     echo "<br><a href='../login_suda/login.html'>Login</a>";
148
149     header('Location: http://localhost:8080/ASIS 3.0/login_suda/login.html');//redirige a la página de login si el
150     usuario quiere ingresar sin iniciar sesion
151
152     exit;
153 }
154
155 $now = time();
156
157 if($now > $_SESSION['expire']) {
158     session_destroy();
159     header("Location: http://localhost:8080/ASIS 3.0/login_suda/login.html");//redirige a la página de login,
160     modifica la url a tu conveniencia
161     echo "Tu sesion a expirado,
162     <a href='login.html'>Inicia Sesion</a>";
163     exit;
164 }
165 }
166 ?>

```

Figura 38. Creación de seguridad para la página.

Fuente: el autor.

Para que la página HTML se conecte a la Base de Datos se necesita crear un archivo de conexión el cual es muy sencillo se agrega la dirección del servidor, un usuario, contraseña y nombre de la Base a la cual se va a conectar.



```

C:\xampp\htdocs\asistencia\conexion.php (asistencia) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
  asistencia
    bd
    css
    ethernet
    fonts
    js
  < Asietncia De Estudiantes.html
  conexion.php
  listar.php

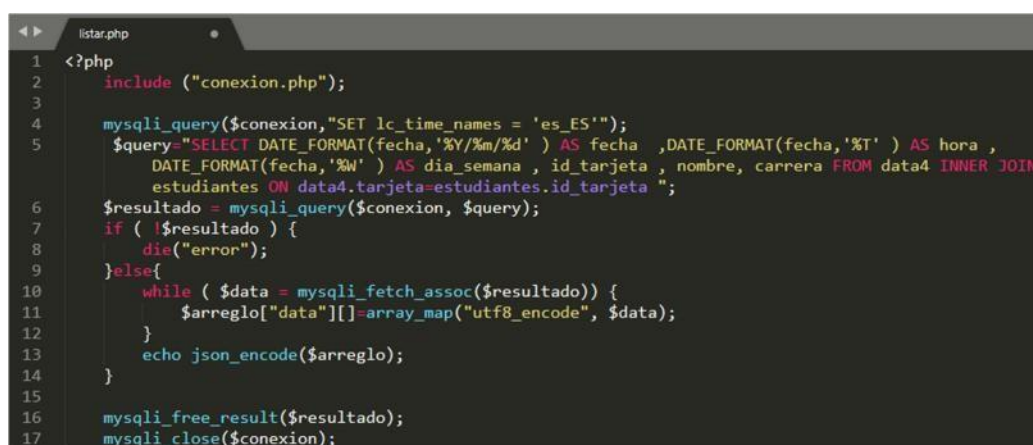
conexion.php
1 <?php
2 $server = "localhost";
3 $user = "arduino";
4 $password = "1234";//poner tu propia contraseña, si tienes una.
5 $bd = "registro_asistencia_alumnos";
6
7 $conexion = mysqli_connect($server, $user, $password, $bd);
8 if (!$conexion){
9     die('Error de Conexión: ' . mysqli_connect_errno());
10 }
11 ?>

```

Figura 39. Programación del archivo conexion.php para poder conectar la página de asistencia de alumnos a la base de datos.

Fuente: el autor.

Para que la página HTML pueda mostrar la información dentro de la Base de Datos es necesario crear un archivo que genere la consulta SQL hacia la Base, primero se instancia el archivo conexion.php, se configura el idioma de los días de la semana ya que estos se encuentran en inglés porque es el idioma por defecto del gestor de Bases de Datos MySQL, se crea una consulta SQL en el cual se selecciona la *fecha*, *hora*, *dia_semana*, *id_tarjeta*, *nombre*, *carrera*, con la condición que si tarjeta de la tabla data es = id_tarjeta de la tabla estudiantes este muestre los registros de asistencia, si hay alguna asistencia y el estudiante no está agregado en la tabla estudiantes estos registros no se mostrarán, finalmente los datos que obtenga la sentencia SQL serán codificados en formato JSON esto para que cualquier persona que sepa hacking e intente capturar o alterar los datos estos los obtenga cifrados así haciendo imposible la lectura o modificación de los mismos, por último se cierra la conexión hacia la base después de haber realizado la consulta.



```

listar.php
1 <?php
2 include ("conexion.php");
3
4 mysqli_query($conexion,"SET lc_time_names = 'es_ES'");
5 $query="SELECT DATE_FORMAT(fecha,'%Y/%m/%d' ) AS fecha ,DATE_FORMAT(fecha,'%T' ) AS hora ,
6 DATE_FORMAT(fecha,'%W' ) AS dia_semana , id_tarjeta , nombre, carrera FROM data4 INNER JOIN
7 estudiantes ON data4.tarjeta=estudiantes.id_tarjeta ";
8 $resultado = mysqli_query($conexion, $query);
9 if ( !$resultado ) {
10     die("error");
11 }else{
12     while ( $data = mysqli_fetch_assoc($resultado) ) {
13         $arreglo["data"][]=array_map("utf8_encode", $data);
14     }
15     echo json_encode($arreglo);
16
17 mysqli_free_result($resultado);
18 mysqli_close($conexion);

```

Figura 40. Programación del archivo listar.php para poder mostrarlos datos dentro de la base de datos en la página asistencia de alumnos.

Fuente: el autor.

ASISTENCIA DEL AULA 402

Fecha	Hora	Día	Id Tarjeta	Nombre	Carrera
2019/05/09	12:05:51	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	12:05:49	jueves	10517522143	erick xavier quishpe andrade	sistemas
2019/05/09	11:53:50	jueves	10517522143	erick xavier quishpe andrade	sistemas
2019/05/09	11:51:33	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	11:51:33	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	11:51:33	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	11:51:32	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	11:51:31	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	11:51:30	jueves	418621443	karen rojas dayana	sistemas
2019/05/09	11:33:48	jueves	10517522143	erick xavier quishpe andrade	sistemas

Mostrar 10 registros

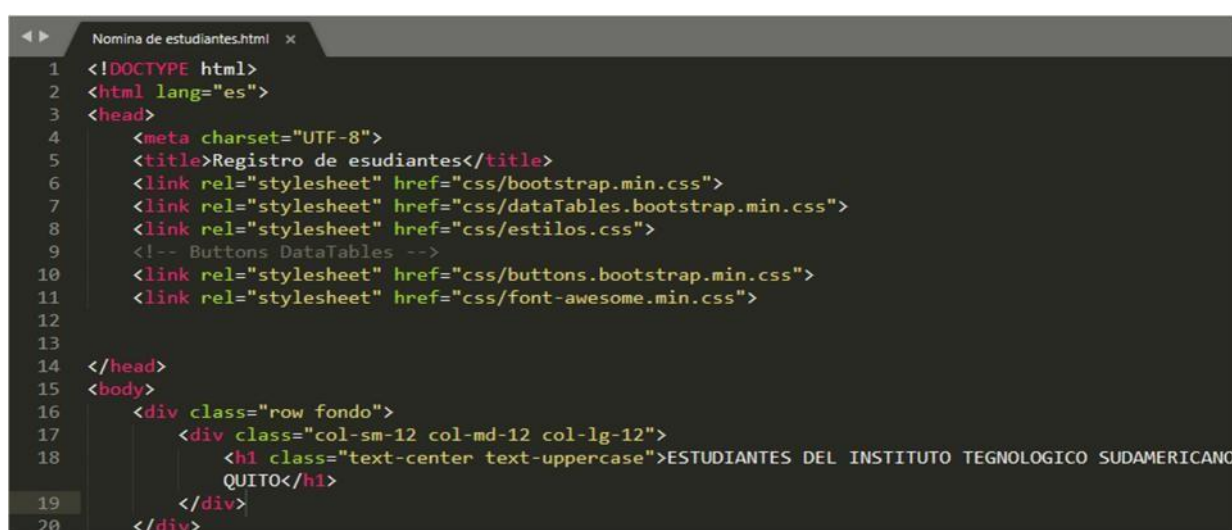
Anterior
1
2
3
4
Siguiente

Figura 41. Funcionamiento de la página de asistencia donde se muestra un ejemplo de asistencia.

Fuente: el autor.

Para poder registrar alumnos nuevos, borrarlos o modificar información se ha creado una nueva página HTML llamada `Nómina de estudiantes.html`. En esta página se ha realizado, mediante programación, un formulario para inscripción de alumnos, agregando funciones de eliminar alumnos o modificarlos esto se lo realiza en otro archivo llamado `guardar.php` donde se encuentran las funciones de guardar, modificar y eliminar estudiantes, también se puede descargar un archivo de la nómina de todos los alumnos que se encuentren registrados si esto fuese necesario.

Primero se da un nombre a la página, se agregan las librerías necesarias, se crea un título para poder identificar la página.



```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Registro de esudiantes</title>
6   <link rel="stylesheet" href="css/bootstrap.min.css">
7   <link rel="stylesheet" href="css/dataTables.bootstrap.min.css">
8   <link rel="stylesheet" href="css/estilos.css">
9   <!-- Buttons DataTables -->
10  <link rel="stylesheet" href="css/buttons.bootstrap.min.css">
11  <link rel="stylesheet" href="css/font-awesome.min.css">
12
13
14 </head>
15 <body>
16   <div class="row fondo">
17     <div class="col-sm-12 col-md-12 col-lg-12">
18       <h1 class="text-center text-uppercase">ESTUDIANTES DEL INSTITUTO TEGNOLOGICO SUDAMERICANO
19         QUITO</h1>
20     </div>
21   </div>

```

Figura 42. Se agregan las librerías necesarias.

Fuente: el autor.

Se crea un formulario para el registro de estudiantes este se lo concatena con la función registrar del archivo `guardar.php` y se le asigna que se lo va a realizar mediante el método POST, se crean etiquetas label para ingresar los datos del estudiante las cuales son *Id Tarjeta*, *Nombres* y *Apellidos*, *Carrera*, toda esta información se guarda y se envía a la función registrar, se crea un botón para guardar y otro para listar los estudiantes existentes.

```

21
22
23 <div class="row">
24   <div id="cuadro2" class="col-sm-12 col-md-12 col-lg-12">
25     <form class="form-horizontal" action="" method="POST">
26       <div class="form-group">
27         <h3 class="col-sm-offset-2 col-sm-8 text-center">
28           Formulario de Registro de Usuarios</h3>
29       </div>
30       <!--<input type="hidden" id="idusuario" name="idusuario" value="0"-->
31       <input type="hidden" id="opcion" name="opcion" value="registrar">
32       <div class="form-group">
33         <label for="id_tarjeta" class="col-sm-2 control-label">Id Tarjeta</label>
34         <div class="col-sm-8"><input id="id_tarjeta" name="id_tarjeta" type="text" class="
35           form-control" autofocus></div>
36       </div>
37       <div class="form-group">
38         <label for="nombre" class="col-sm-2 control-label">Nombres y apellidos</label>
39         <div class="col-sm-8"><input id="nombre" name="nombre" type="text" class="
40           form-control" ></div>
41       </div>
42       <div class="form-group">
43         <label for="carrera" class="col-sm-2 control-label">Carrera</label>
44         <div class="col-sm-8"><input id="carrera" name="carrera" type="text" class="
45           form-control" ></div>
46       </div>
47       <div class="form-group">
48         <div class="col-sm-offset-2 col-sm-8">
49           <input id="" type="submit" class="btn btn-primary" value="Guardar">
50           <input id="btn_listar" type="button" class="btn btn-primary" value="Listar">
51         </div>
52       </div>
53     </form>
54   </div>
55 </div>

```

Figura 43. Formulario de registro de nuevos estudiantes.

Fuente: el autor.

Se crea la tabla *dt_cliente*, dentro de esta se crea los encabezados para las columnas los cuales se llamarán *Id Tarjeta*, *Nombre*, *Carrera*, se crea un encabezado más sin nombre para que aquí vayan los botones de eliminar y modificar estudiantes.

```

49 </form>
50 <div class="col-sm-offset-2 col-sm-8">
51   <p class="mensaje"></p>
52 </div>
53
54 </div>
55 </div>
56 <div class="row">
57   <div id="cuadro1" class="col-sm-12 col-md-12 col-lg-12">
58     <div class="col-sm-offset-2 col-sm-8">
59       <h3 class="text-center"> <small class="mensaje"></small></h3>
60     </div>
61     <div class="table-responsive col-sm-12">
62       <table id="dt_cliente" class="table table-bordered table-hover" cellspacing="0" width="
63         100%">
64         <thead>
65           <tr>
66             <th>Id Tarjeta</th>
67             <th>Nombre</th>
68             <th>Carrera</th>
69             <th></th>
70           </tr>
71         </thead>
72       </table>
73     </div>
74   </div>
75 </div>

```

Figura 44. Creación de datatable y nombres de las columnas.

Fuente: el autor.

Se crea un formulario para eliminar estudiantes, se le asigna que se lo va realizar mediante el método POST, se elige el *id_tarjeta* de la tabla y se elimina todos los datos que contenga este, para esto se crea un botón con un mensaje de eliminar usuario, cuando se da clic aparece una ventana la cual nos muestra un mensaje de si se está seguro de que se va a eliminar el usuario con dos botones uno de aceptar y el otro de cancelar.

```

76 <div>
77 <form id="frmEliminarUsuario" action="" method="POST">
78 <input type="hidden" id="id_tarjeta" name="id_tarjeta" value="">
79 <input type="hidden" id="opcion" name="opcion" value="eliminar">
80 <!-- Modal -->
81 <div class="modal fade" id="modalEliminar" role="dialog" aria-labelledby="modalEliminarLabel">
82 <div class="modal-dialog" role="document">
83 <div class="modal-content">
84 <div class="modal-header">
85 <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
86
87 <h4 class="modal-title" id="modalEliminarLabel">Eliminar Usuario</h4>
88 </div>
89
90 <div class="modal-body">
91 ¿Está seguro de eliminar al usuario?<strong data-name=""></strong>
92 </div>
93 <div class="modal-footer">
94 <button type="button" id="eliminar-usuario" class="btn btn-primary" data-dismiss="modal">Aceptar</button>
95 <button type="button" class="btn btn-default" data-dismiss="modal">Cancelar</button>
96 </div>
97 </div>
98 </div>
99 </div>
100 <!-- Modal -->
101 </form>
102 </div>

```

Figura 45. Formulario para eliminación de estudiantes.

Fuente: el autor.

Se continúa agregando las librerías de lenguaje JavaScript donde están los botones para exportar a Excel y PDF , agregar botones a la tabla para las funciones de eliminar y editar usuarios, se sigue con la función de lectura de los datos se crea la función *listar ()*, *guardar ()*, *eliminar ()*, la función listar se crea para poder realizar la publicación de los datos capturados por el archivo listar.php estos datos son capturados mediante AJAX (captura de datos de un servidor) y son puestos en la tabla mediante el método POST (envío de datos de forma invisible para el cliente), se crea la función guardar para el momento de agregar nuevos estudiantes se los pueda guardar en la Base de Datos, la función eliminar para poderlos eliminar.

```

103
104 <script src="js/jquery-1.12.3.js"></script>
105 <script src="js/bootstrap.min.js"></script>
106 <script src="js/jquery.dataTables.min.js"></script>
107 <script src="js/dataTables.bootstrap.js"></script>
108 <!--botones DataTables-->
109 <script src="js/dataTables.buttons.min.js"></script>
110 <script src="js/buttons.bootstrap.min.js"></script>
111 <!--Libreria para exportar Excel-->
112 <script src="js/jzip.min.js"></script>
113 <!--Librerias para exportar PDF-->
114 <script src="js/pdfmake.min.js"></script>
115 <script src="js/vfs_fonts.js"></script>
116 <!--Librerias para botones de exportación-->
117 <script src="js/buttons.html5.min.js"></script>
118
119 <script>
120     $(document).on("ready", function(){
121
122         listar();
123         guardar();
124         eliminar();
125
126     });
127
128     $("#btn_listar").on("click",function(){
129         listar();
130     });

```

Figura 46. Declaración de librerías para botones de datatable y lectura de funciones listar (), guardar (), eliminar ().

Fuente: el autor.

Se crea la función guardar, dentro de esta se crea con la función keypress para que solamente al dar clic se realice la función guardar, al dar clic se realiza la función AJAX especificando el método POST como la extracción de los datos, se indica que esta función se la va realizar en el archivo guardar.php y que los datos se los va extraer del formulario form, estos datos se los transforma en tipo JSON para poder enviarlos al servidor, se muestra los mensajes de confirmación si se ha realizado o no con éxito la función, se limpia los datos dentro del formulario form y se lista todos los datos dentro de la Base de Datos.


```

131
132     var guardar=function(){
133         $("form").keypress(function(e) {
134             if (e.which == 13) {
135                 return false;
136             }
137         });
138         $("form").on("submit",function(e){
139             e.preventDefault();
140             var frm=$(this).serialize();
141
142             $.ajax({
143                 method:"POST",
144                 url: "guardar.php",
145                 data: frm
146             }).done( function( info ){
147                 var json_info=JSON.parse(info);
148                 mostrar_mensaje(json_info);
149                 limpiar_datos();
150                 listar();
151             });
152         });
153     });
154 }

```

Figura 47. Creación de la función guardar ().

Fuente: el autor.

Se crea la función eliminar la cual al dar clic en el botón eliminar toma el dato de id_tarjeta y mediante la función AJAX envía el dato hacia función eliminar dentro del archivo guardar.php, después procede a listar los datos, transformarlos en tipo JSON, muestra el mensaje si se ha borrado con éxito el estudiante y limpia los datos dentro del formulario.

```

155     var eliminar = function(){
156         $("#eliminar-usuario").on("click", function(){
157             var id_tarjeta=$("#frmEliminarUsuario #id_tarjeta").val();
158             var opcion = $("#frmEliminarUsuario #opcion").val();
159             $.ajax({
160                 method:"POST",
161                 url: "guardar.php",
162                 data: {"id_tarjeta":id_tarjeta,"opcion":opcion}
163             }).done( function( info ){
164                 listar();
165                 var json_info=JSON.parse(info);
166                 mostrar_mensaje(json_info);
167                 limpiar_datos();
168             });
169         });
170     });
171 }

```

Figura 48. Creación de la función eliminar ().

Fuente: el autor.

Se crea la función mostrar_mensaje, con esta función se muestran todos los mensajes necesarios para los formularios dependiendo de la función de guardar, modificar y eliminar, y la respuesta que haya sido necesaria, al final estas respuestas son mostradas en formato HTML para hacerlas visibles al usuario.

```

172
173
174     var mostrar_mensaje = function( informacion ){
175     var texto = "", color = "";
176     if( informacion.respuesta == "BIEN" ){
177         texto = "<strong>Bien!</strong> Se han guardado los cambios correctamente.";
178         color = "#379911";
179     }else if( informacion.respuesta == "ERROR"){
180         texto = "<strong>Error</strong>, no se ejecutó la consulta.";
181         color = "#C9302C";
182     }else if( informacion.respuesta == "EXISTE" ){
183         texto = "<strong>Información!</strong> el usuario ya existe.";
184         color = "#5b94c5";
185     }else if( informacion.respuesta == "VACIO" ){
186         texto = "<strong>Advertencia!</strong> debe llenar todos los campos solicitados.";
187         color = "#ddb11d";
188     }else if( informacion.respuesta == "OPCION_VACIA" ){
189         texto = "<strong>Advertencia!</strong> la opción no existe o esta vacia, recargar la
190             página.";
191         color = "#ddb11d";
192     }
193     $(".mensaje").html( texto ).css({"color": color });
194     $(".mensaje").fadeOut(5000, function(){
195         $(".mensaje").html("");
196         $(".mensaje").fadeIn(3000);
197     });
198 }

```

Figura 49. Creación de la función mostrar_mensaje ().

Fuente: el autor.

Se crea la función limpiar_datos, para limpiar las etiquetas label cada vez que se realice la función de guardar o modificar los datos.

Se crea la función listar para mostrar todos los estudiantes que se encuentran en la Base de Datos registrados mediante la función AJAX se crea las columnas *id_tarjeta*, *nombre*, *carrera* y por último la columna que contendrá los botones de modificar y eliminar estudiantes.

```

199     var limpiar_datos = function(){
200     $("#opcion").val("registrar");
201     $("#id_tarjeta").val("").focus();
202     $("#nombre").val("");
203     $("#carrera").val("");
204
205     }
206
207     var listar = function(){
208     $("#cuadro2").slideUp("slow");
209     $("#cuadro1").slideDown("slow");
210     var table = $("#dt_cliente").DataTable({
211         "destroy":true,
212         "ajax":{
213             "method":"POST",
214             "url":"listar.php"
215         },
216         "columns":[
217             {"data":'id_tarjeta',"width": "10%"},
218             {"data":'nombre',"width": "10%"},
219             {"data":'carrera',"width": "10%"},
220             {"defaultContent":"<button type='button' class='editar btn btn-primary'><i class='fa fa-pencil-square-o'></i></button> <button type='button' class='eliminar btn btn-danger' data-toggle='modal' data-target='#modalEliminar' ><i class='fa fa-trash-o'></i></button>","width": "10%"}
221
222     ],

```

Figura 50. Creación de la función listar ().

Fuente: el autor.

Se crea la función *idioma_espanol*, la cual es una función que ya está programada solo hay que buscar en la página oficial de jQuery y agregarlo al código para que la tabla que por defecto está en idioma inglés lo transforme a español.

Se crea los botones de agregar nuevos estudiantes, exportar en EXCEL y PDF.

```

223
224     "language":idioma_espanol,
225     "dom":"<'row'<'form-inline' <'col-sm-offset-5'B>>>"
226     + "<'row' <'form-inline' <'col-sm-1'f>>>"
227     + "<rt>"
228     + "<'row'<'form-inline'"
229     + " <'col-sm-6 col-md-6 col-lg-6'l'"
230     + "<'col-sm-6 col-md-6 col-lg-6'p>>>",
231     "buttons":[
232     {
233         "text":"<i class='fa fa-user-plus'></i>",
234         "titleAttr":"Agregar Usuario",
235         "className": "btn btn-success",
236         "action": function(){
237             agregar_nuevo_usuario();
238         }
239     },
240     {
241         extend: 'excelHtml5',
242         text: '<i class="fa fa-file-excel-o"></i>',
243         titleAttr: 'Excel'
244     },
245     {
246         extend: 'pdfHtml5',
247         text: '<i class="fa fa-file-pdf-o"></i>',
248         titleAttr: 'PDF'
249     }
250 ]
251 });
252 obtener_data_editar("#dt_cliente tbody",table);
253 obtener_id_eliminar("#dt_cliente tbody",table);
254
255 }

```

Figura 51. Elección de idioma español de la tabla, botón de agregar usuarios nuevos.

Fuente: el autor.

Se crea la función *agregar_nuevo_usuario*, en esta se lista la función *limpiar_datos* y se agrega el deslizamiento de la página hacia arriba y abajo.

Se crea la función *obtener_data_editar*, dentro de esta se agrega que al dar clic en el botón este obtiene la *id_tarjeta*, *nombre*, *carrera* de la fila y lo envía hacia la función modificar dentro del archivo guardar.php.

Se crea la función *obtener_id_eliminar*, al dar clic sobre el botón eliminar este obtiene el *id_tarjeta* y lo envía al formulario de eliminar estudiantes.

```

256
257     var agregar_nuevo_usuario = function(){
258         limpiar_datos();
259         $("#cuadro2").slideDown("slow");
260         $("#cuadro1").slideUp("slow");
261     }
262
263     var obtener_data_editar=function(tbody,table){
264         $(tbody).on("click", "button.editar",function(){
265             var data = table.row( $(this).parents("tr") ).data();
266             id_tarjeta = $("#id_tarjeta").val( data.id_tarjeta),
267             nombre = $("#nombre").val( data.nombre ),
268             carrera = $("#carrera").val( data.carrera ),
269             opcion = $("#opcion").val("modificar");
270             $("#cuadro2").slideDown("slow");
271             $("#cuadro1").slideUp("slow");
272
273         });
274     }
275
276
277     var obtener_id_eliminar = function(tbody, table){
278         $(tbody).on("click", "button.eliminar", function(){
279             var data = table.row( $(this).parents("tr") ).data();
280             var id_tarjeta = $("#frmEliminarUsuario #id_tarjeta").val( data.id_tarjeta );
281         });
282     }

```

Figura 52. Creación de las funciones `agregar_nuevo_usuario ()`, `obtener_data_editar ()`, `obtener_id_eliminar ()`.

Fuente: el autor.

Para agregar el idioma español a la tabla se obtiene una función creada previamente por los desarrolladores de jQuery la cual solo hay que descargarla y agregarla al código.

```

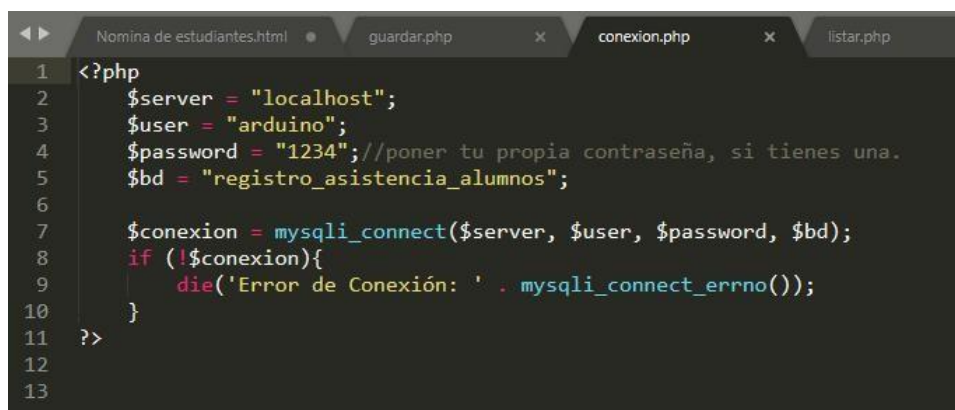
283     |   var idioma_espanol={
284         |       "sProcessing":      "Procesando...",
285         |       "sLengthMenu":     "Mostrar _MENU_ registros",
286         |       "sZeroRecords":    "No se encontraron resultados",
287         |       "sEmptyTable":     "Ningún dato disponible en esta tabla",
288         |       "sInfo":           "Mostrando registros del _START_ al _END_ de un total de _TOTAL_ registros",
289         |       "sInfoEmpty":      "Mostrando registros del 0 al 0 de un total de 0 registros",
290         |       "sInfoFiltered":   "(filtrado de un total de _MAX_ registros)",
291         |       "sInfoPostFix":    "",
292         |       "sSearch":         "Buscar:",
293         |       "sUrl":            "",
294         |       "sInfoThousands":  ",",
295         |       "sLoadingRecords": "Cargando...",
296         |       "oPaging": {
297             |           "sFirst":      "Primero",
298             |           "sLast":       "Último",
299             |           "sNext":       "Siguiente",
300             |           "sPrevious":   "Anterior"
301         |       },
302         |       "oAria": {
303             |           "sSortAscending": ": Activar para ordenar la columna de manera ascendente",
304             |           "sSortDescending": ": Activar para ordenar la columna de manera descendente"
305         |       }
306     |   }
307
308
309     </script>
310 </body>
311 </html>

```

Figura 53. Creación de la función `idioma_espanol ()`.

Fuente: el autor.

Para que la página HTML de registro de alumnos se pueda conectar a la Base de Datos es necesario crear un archivo de conexión hacia la Base.



```

1 <?php
2 $server = "localhost";
3 $user = "arduino";
4 $password = "1234";//poner tu propia contraseña, si tienes una.
5 $bd = "registro_asistencia_alumnos";
6
7 $conexion = mysqli_connect($server, $user, $password, $bd);
8 if (!$conexion){
9     die('Error de Conexión: ' . mysqli_connect_errno());
10 }
11 ?>
12
13

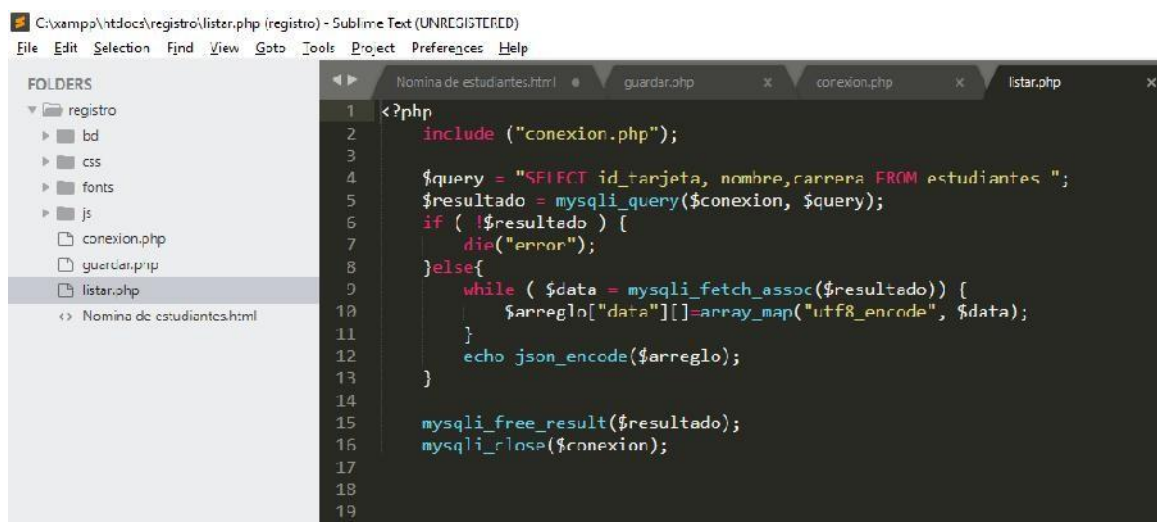
```

Figura 54. Creación del archivo conexión.php de la página registro de alumnos.

Fuente: el autor.

Para poder mostrar los datos se realizará una consulta SQL a la Base de Datos esto se lo realizará en un archivo llamado listar.php, en este archivo se instancia el archivo conexión.php, se realiza con consulta SQL con los datos de la tabla estudiantes, estos se los almacena en una matriz y se los decodifica para poder mostrarlos en la tabla.

Por último, se cierra la conexión hacia la Base de Datos.



```

1 <?php
2 include ("conexion.php");
3
4 $query = "SELECT id_tarjeta, nombre,carrera FROM estudiantes ";
5 $resultado = mysqli_query($conexion, $query);
6 if ( !$resultado ) {
7     die("error");
8 }else{
9     while ( $data = mysqli_fetch_assoc($resultado) ) {
10         $arreglo["data"][]=array_map("utf8_encode", $data);
11     }
12     echo json_encode($arreglo);
13 }
14
15 mysqli_free_result($resultado);
16 mysqli_close($conexion);
17
18
19

```

Figura 55. Programación del archivo listar.php de la página registro de alumnos para poder mostrar los alumnos registrados dentro de la Base de Datos.

Fuente: el autor.

Para las funciones de agregar, eliminar y modificar alumnos se crea un archivo llamado guardar.php se instancia el archivo conexión.php, se crea variables para el almacenamiento de los datos dentro de la tabla.

Se crea un switch para comparar todas las funciones dentro con la variable *\$opcion* para cuando se realice alguna función este venga directo hacia la función necesaria dentro de estas está la función *registrar*, *modificar* y *eliminar*, se utiliza un break al final de cada case para terminar la función ahí y no se vaya a ninguna otra parte del programa que no es necesario.

Antes de ir a la función registrar este compara si hay información dentro del formulario de registro de no haberla envía un mensaje hacia el usuario diciendo que los campos están vacíos.

```

guardar.php
1 <?php
2 include("conexion.php");
3 $id_tarjeta=$_POST["id_tarjeta"];
4 $nombre=$_POST["nombre"];
5 $carrera=$_POST["carrera"];
6 $opcion=$_POST["opcion"];
7 $informacion=[];
8
9 switch ($opcion) {
10     case 'registrar':
11
12         if( $id_tarjeta != "" && $nombre != "" && $carrera != "" ){
13             $existe = existe_usuario($id_tarjeta, $conexion);
14             if ($existe > 0) {
15                 $informacion["respuesta"]="Existe";
16                 echo json_encode($informacion);
17                 # code...
18             }else{
19                 registrar($nombre, $id_tarjeta, $carrera, $conexion);
20             }
21
22         }else{
23             $informacion["respuesta"] = "VACIO";
24             echo json_encode($informacion);
25         }
26         break;
27     case 'modificar':
28         modificar( $id_tarjeta, $nombre,$carrera, $conexion);
29
30         break;
31
32     case 'eliminar':
33         eliminar($id_tarjeta, $conexion);
34         break;

```

Figura 56. Creación de switch para elección si se desea registrar, modificar o eliminar alumnos.

Fuente: el autor.

Función *existe_usuario*, compara que la *id_tarjeta* sea única dentro de la tabla estudiantes de la Base de Datos si lo es envía un mensaje de que existe un usuario con esa identificación.

Función *registrar*, realiza una consulta SQL que registra el usuario con todos los datos obtenidos del formulario.

Función *modificar* envía los datos modificados o actualizados hacia la Base siempre y cuando el *id_tarjeta* sea el mismo este no se puede modificar.

```

guardar.php
35         default:
36             $informacion["respuesta"] = "OPCION VACIA";
37             echo json_encode($informacion);
38             break;
39
40     }
41     function existe_usuario($id_tarjeta, $conexion){
42         $query = "SELECT id_tarjeta FROM estudiantes WHERE id_tarjeta = '$id_tarjeta'";
43         $resultado = mysqli_query($conexion, $query);
44         $existe_usuario = mysqli_num_rows( $resultado );
45         return $existe_usuario;
46     }
47
48     function registrar($nombre, $id_tarjeta, $carrera, $conexion){
49         $query = "INSERT INTO estudiantes VALUES( '$id_tarjeta', '$nombre', '$carrera' )";
50         $resultado = mysqli_query($conexion, $query);
51         verificar_resultado($resultado);
52         cerrar($conexion);
53     }
54
55
56     function modificar( $id_tarjeta, $nombre,$carrera, $conexion){
57         $query="UPDATE estudiantes SET nombre='$nombre', carrera='$carrera'
58         WHERE id_tarjeta='$id_tarjeta' " ;
59
60         $resultado = mysqli_query($conexion, $query);
61         verificar_resultado( $resultado );
62         cerrar( $conexion );
63     }

```

Figura 57. Creación de funciones existe_usuario, registrar y modificar.

Fuente: el autor.

Función *eliminar* realiza una consulta SQL en la cual elimina los datos dentro de la Base que contengan el *id_tarjeta* que selecciona.

Función *verificar_resultado* envía un mensaje de error o bien a la función mostrar mensaje para indicar que el estudiante ha sido registrado exitosamente o hay algún error.

Función *cerrar* termina la conexión con la Base de Datos.

```

guardar.php
64
65     function eliminar($id_tarjeta, $conexion){
66         $query = "DELETE FROM estudiantes WHERE id_tarjeta='$id_tarjeta'";
67         $resultado = mysqli_query($conexion, $query);
68         verificar_resultado( $resultado );
69         cerrar( $conexion );
70     }
71     function verificar_resultado($resultado){
72         if(!$resultado) $informacion["respuesta"]="ERROR";
73         else $informacion["respuesta"]="BIEN";
74         echo json_encode($informacion);
75     }
76     function cerrar($conexion){
77         mysqli_close($conexion);
78     }
79
80     >>

```

Figura 58. Creación de funciones eliminar, verificar_resultado y cerrar.

Fuente: el autor.

ESTUDIANTES DEL INSTITUTO TECNOLÓGICO SUDAMERICANO QUITO

Buscar:





Id Tarjeta	Nombre	Carrera		
418621443	karen rojas dayana	sistemas	 	
1234567890	kevin andrango	sistemas	 	
10289101172	junior rojas	sistemas	 	
10517522143	erick xavier quishpe andrade	sistemas	 	

Mostrar registros

Figura 59. Funcionamiento de la página de registro de alumnos a la Base de Datos.

Fuente: el autor.

ESTUDIANTES DEL INSTITUTO TECNOLÓGICO SUDAMERICANO QUITO

Formulario de Registro de Usuarios

Id Tarjeta

Nombres y apellidos

Carrera

Figura 60. Formulario para el registro de nuevos estudiantes.

Fuente: el autor.

ESTUDIANTES DEL INSTITUTO TECNOLÓGICO SUDAMERICANO QUITO

Formulario de Registro de Usuarios

Id Tarjeta

Nombres y apellidos

Carrera

Figura 61. Cambio de datos de un estudiante.

Fuente: el autor.

Se crea una página de inicio para que solamente personal docente pueda ingresar al registro de asistencia de los estudiantes, así no habrá problema con que los estudiantes puedan realizar daños al sistema o que cualquier otra persona intente plagiar la información que se encuentra dentro de la Base de Datos o haya alguna fuga de información.



Figura 62. Personal que interactúa con la Base de Datos.

Fuente: el autor.

Figura 63. Página de inicio del Sistema de Registro de Asistencia.

Fuente: el autor.

Para la página de login se crea que al momento de dar clic en el botón login este capture el nombre del usuario y el tiempo en ese instante y esto lo enviará a checklogin.php, por último, se crean etiquetas label para ingresar el Usuario y contraseña y el botón de login.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1">
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6 <link rel="stylesheet" href="css/estilos.css">
7 <style>
8 body {font-family: Arial, Helvetica, sans-serif; background-image: url("fondo2.jpg"); }
9 </style>
10 </head>
11 <body>
12 <div id="id01" class="modal">
13
14 <form class="modal-content animate" action="checklogin.php" method="post">
15 <div class="imgcontainer">
16 <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
17 
18 </div>
19
20 <div class="container">
21 <label for="uname"><b>Usuario</b></label>
22 <input type="text" placeholder="Ingresar Usuario" name="username" required>
23
24 <label for="psw"><b>Contraseña</b></label>
25 <input type="password" placeholder="Ingresa tu Contraseña" name="password" required>
26
27 <button type="submit">Login</button>
28
29 </div>
30
31 </form>
32 </div>
33

```

Figura 64. Página login.html.

Fuente: el autor.

Se crea un archivo que compara si el usuario y contraseña existen en la Base de Datos llamado ceklogin.php, primero se incluye el archivo de conexión.php, realiza una nueva conexión hacia la base de datos, si la conexión falla este enviará un mensaje de error, se continúa obteniendo el nombre de usuario y contraseña, se realiza una consulta SQL si el nombre de usuario es correcto este continúa verificando si la contraseña es correcta, si lo es, inicializa la variable *loggedin* captura el nombre del usuario ,inicializa la variable *start* la cual controla el tiempo que el usuario puede estar sin utilizar el sistema, se inicializa la variable *expire* la cual controla que la sesión puede estar iniciada solo 5 minutos después no utilizarla, por último te redirecciona a la página de menú donde se encuentran todas las Aulas y el Registro de Estudiantes, si el usuario o contraseña no son correctos este muestra un mensaje de error y pide que se lo vuelva a intentar.

```

4  <?php
5  include 'conexion.php';
6  $conexion = new mysqli($host_db, $user_db, $pass_db, $db_name);
7  if ($conexion->connect_error) {
8      die("La conexion falló: " . $conexion->connect_error);
9  }
10 $username = $_POST['username'];
11 $password = $_POST['password'];
12
13 $sql = "SELECT * FROM $tbl_name WHERE nombre_usuario = '$username'";
14 $result = $conexion->query($sql);
15 if ($result->num_rows > 0) {
16     $row = $result->fetch_array(MYSQLI_ASSOC);
17     // if (password_verify($password, $row['password'])) {
18     if ($password==$row['password']) {
19         $_SESSION['loggedin'] = true;
20         $_SESSION['username'] = $username;
21         $_SESSION['start'] = time();
22         $_SESSION['expire'] = $_SESSION['start'] + (5 * 60);
23
24         echo "Bienvenido! " . $_SESSION['username'];
25         echo "<br><br><a href=redireccion.html>Panel de Control</a>";
26         header('Location: http://localhost:8080/ASIS 3.0/login_suda/redireccion.php');
27         //redirecciona a la pagina del usuario
28     } else {
29         echo "Username o Password estan incorrectos.";
30         echo "<br><a href='login.html'>Volver a Intentarlo</a>";
31     }
32     mysqli_close($conexion);
33 }>

```

Figura 65. Creación del archivo checklogin.php.

Fuente: el autor.

Se crea una página de menú que muestra las aulas en las que se está registrando la asistencia para poder realizar una fácil búsqueda de datos, también muestra la página de registro de estudiantes nuevos, se puede ingresar dando clic.



Figura 66. Menú que muestra la asistencia por aulas y registro de estudiantes nuevos.

Fuente: el autor.

Se crea un archivo de conexión hacia la Base de Datos de los usuarios.

```
conexion.php x redirec
1 |
2 <?php
3 $host_db = "localhost";
4 $user_db = "arduino";
5 $pass_db = "1234";
6 $db_name = "usuariosDB";
7 $tbl_name = "usuarios";
8 ?>
```

Figura 67. Archivo de conexión hacia la Base de Datos de los usuarios.

Fuente: el autor.

Para poder ingresar a la página de redirección primero realiza una comprobación de si hay un usuario conectado de no haberlo este te re dirige hacia la página del Inicio.

```

redireccion.php
1  <?php
2  session_start();
3  if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true) {
4
5  } else {
6      echo "Inicia Sesion para acceder a este contenido.<br>";
7      header('Location: http://localhost:8080/ASIS 3.0/login_suda/login.html');
8      //redirige a la página de login si el usuario quiere ingresar sin iniciar sesion
9      exit;
10 }
11 $now = time();
12
13 if($now > $_SESSION['expire']) {
14     session_destroy();
15     header('Location: http://localhost:8080/login/login.html');
16     //redirige a la página de login, modifica la url a tu conveniencia
17     echo "Tu sesion a expirado,
18     <a href='login.html'>Inicia Sesion</a>";
19     exit;
20 }
21 ?>

```

Figura 68. Creación de la seguridad para el archivo de redirección.php.

Fuente: el autor.

Se crea un título para la página de menú y se da estilos a los botones, se crean 5 botones en los cuales al momento de dar clic encima de ellos estos te re direccionan hacia la pestaña que se desee.

```

redireccion.php
22
23 <!DOCTYPE html >
24 <html lang="es">
25 <head>
26 <title>RAE SUDA</title>
27 <style type="text/css">
28     .boton_1{
29         text-decoration: none; padding: 3px; padding-left: 10px; padding-right: 10px;
30         font-family: helvetica;font-weight: 300; font-size: 50px; font-style: italic;
31         color: blue; background-color: black; border-radius: 15px; border: 3px double Blue;
32     }
33     .boton_1:hover{
34         opacity: 0.6;
35         text-decoration: none;
36     }
37 </style>
38 </head>
39 <body background="fondo2.jpg">
40 <center>
41 <a >>/a><br>
42 <a class="boton_1" href=" ../registro/Nomina%20de%20estudiantes.html " name="boton_1">Registro de Estudiantes</a><br>
43 <a >>/a><br>
44 <a class="boton_1" href=" ../asistencia_301/Asistencia%20De%20Estudiantes.php" name="boton_301" >Aula 301</a><br>
45 <a >>/a><br>
46 <a class="boton_1" href=" ../asistencia_401/Asistencia%20De%20Estudiantes.php " name="boton_401 " >Aula 401</a><br>
47 <a >>/a><br>
48 <a class="boton_1" href=" ../asistencia_402/Asistencia%20De%20Estudiantes.php " >Aula 402</a><br>
49 <a >>/a><br>
50
51 <a class="boton_1" href=" ../asistencia_403/Asistencia%20De%20Estudiantes.php " >Aula 403</a><br>
52 </center>
53 </form>
54 </body>
55 </html>

```

Figura 69. Creación del del menú de Aulas y Registro de Estudiantes.

Fuente: el autor.

6.3.2 Creación de filtro de asistencia en EXCEL

Se crea en EXCEL un archivo llamado FiltroDatos habilitado para macros, dentro de este archivo se crea un filtro para la asistencia de estudiantes dependiendo de cómo el profesor desee obtener la asistencia del estudiante ya sea por una fecha, hora, día, Id Tarjeta, Nombre, Carrera en específico o también si lo desea puede obtenerla en un rango de Fecha y Hora.



Figura 70. Filtro de asistencia de Estudiantes en EXCEL.

Fuente: el autor.

Para poder filtrar se ha utilizado la función AdvanceFilter la cual funciona realizando una búsqueda en un rango de datos en este caso el rango de datos está en la hoja llamada *Data* y esta la filtra por los criterios especificados y los copia en un rango deseado para poder filtrar en un rango de Fecha y Hora se ha creado otro filtro con los resultados del primero y así poder mostrar los datos deseados dentro de un rango.

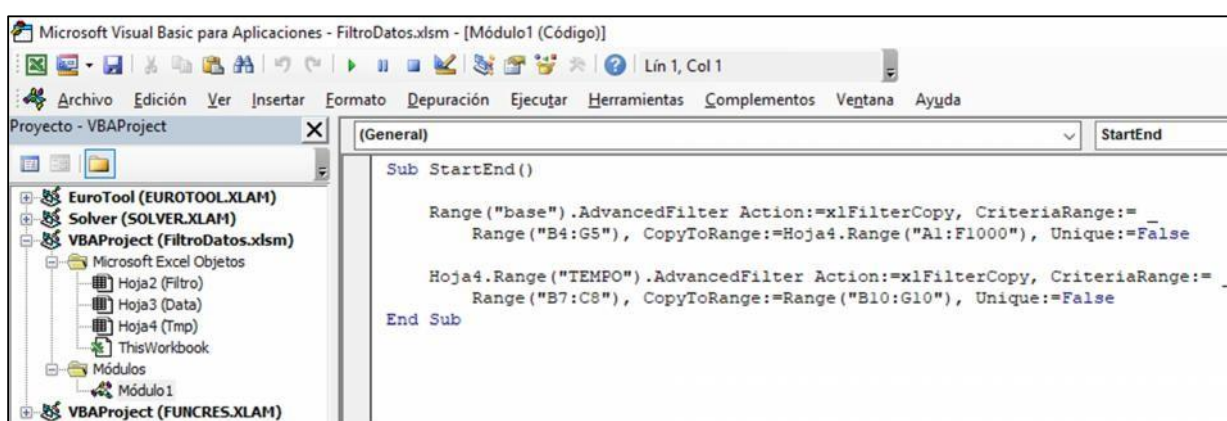


Figura 71. Programación para filtro de datos en EXCEL.

Fuente: el autor.

Para la hoja Data se realiza una consulta desde el archivo que se descarga en EXCEL de la página de Asistencia de Estudiantes y este lo actualiza cada vez que se abra el archivo de FiltroDatos, con esto la Base se ir actualizando constantemente sin necesidad de estar copiando los datos cada vez que sean necesarios.

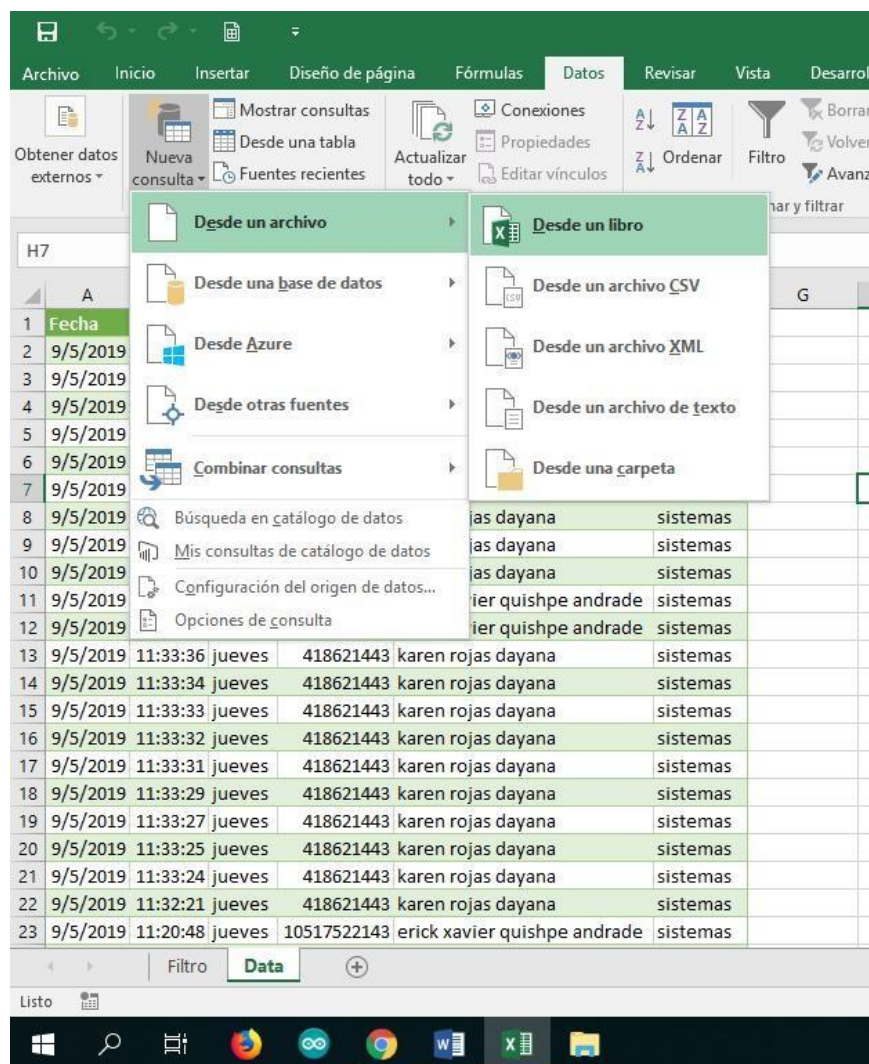


Figura 72. Consulta hacia archivo EXCEL que contiene la Base de Datos de la asistencia.

Fuente: el autor.

6.4 Pruebas de correcto funcionamiento

Se realizaron pruebas de funcionamiento durante un mes en el cual se fueron resolviendo problemas de conectividad a la red, fallos de lectores RFID, hasta llegar al punto óptimo de funcionamiento de los dispositivos de envío de información.

Dispositivo hardware	Conexión a la red institucional	Lectura de Tarjetas	Conexión con el servidor	Conexión Interfaz Gráfica	Nro. de pruebas iniciales realizadas	Nro. de pruebas finales realizadas
Aula 301	Si	Si	Si	Si	10/10	10/10
Aula 401	Si	Si	Si	Si	9/10	10/10
Aula 402	Si	Si	Si	Si	6/10	10/10
Aula 403	Si	Si	Si	Si	7/10	10/10

Figura 73. Tabla comparativa de pruebas realizadas.

Fuente: el autor.

Se solucionó el problema de conectividad de la Base de Datos en la red interna del Instituto.



Figura 74. Funcionamiento y ubicación del servidor en el 4to piso del edificio matriz de la institución.

Fuente: el autor.

Cuando no hay ningún problema de conexión se mantiene encendido el led de color verde y hay un pitido corto para mostrar que no hay errores y se ha registrado correctamente la asistencia.

De no haber ningún problema de conexión hacia la red cuando se lo conecta a la fuente de energía este enciende su led de color verde y lo mantiene así todo el tiempo de no haber problemas o fallas.



Figura 75. Prueba de registro de asistencia.

Fuente: el autor.

Cuando hay un error de conexión hacia la red este enciende los leds al mismo tiempo y los mantiene así hasta solucionar el problema de conexión con la red.



Figura 76. Error de conexión hacia la red.

Fuente: el autor.

Cuando se intenta registrar una asistencia y el servidor por algún motivo se encuentre apagado o haya problema de conexión se encenderá el led de color rojo indicando que hay error en la

7. Conclusiones y Recomendaciones

7.1 Conclusiones

- Se definió los requerimientos del sistema los cuales fueron crear un dispositivo que envíe la información de cada estudiante a una Base de Datos ubicada en un servidor en el cuarto piso y las tecnologías disponibles para realizar este dispositivo fueron con la plataforma Arduino y tecnología Web que permite que la información se vea reflejada en una interfaz gráfica diseñada en HTML.
- Se diseñó el sistema de registro de asistencia desarrollando un esquema de funcionamiento del mismo.
- Se implementó el hardware que conforma el sistema en base a Arduino que es el cerebro del dispositivo de envío de información hacia la Base de Datos.
- Se desarrolló una Base de Datos en MySQL dentro del servidor local en el cual contiene el registro de asistencia de los estudiantes de la carrera de Sistemas de automatización y sus datos necesarios para mostrar en la interfaz gráfica.
- Se implementó el sistema con la instalación de los dispositivos de lectura de tarjetas y envío de información en los laboratorios de Informática 301, 401, 402 y 403.
- Se realizó las pruebas de funcionamiento del sistema probando los dispositivos de envío de información y el correcto funcionamiento de la Base de Datos.
- Se diseñó e implementó un sistema de registro de asistencia a clases de los estudiantes del Instituto Superior Tecnológico Sudamericano Quito llamado RAE-SUDA.
- El manejo de este sistema es muy sencillo intuitivo por lo cual cualquier persona sin ningún conocimiento extenso de este lo puede utilizar sin ningún problema.

7.2 Recomendaciones

- Se recomienda que, con la finalidad de mejorar el sistema, crear un contabilizador de asistencia automatizado dentro de la interfaz gráfica.
- Se recomienda realizar mantenimientos a los dispositivos de envío de información, cableado y correcto funcionamiento de conexión a la red, esto se lo deberá realizar trimestralmente.
- Los dispositivos pueden ser instalados en las aulas de todo el Instituto a disposición y con cargo de gastos a las autoridades responsables de la Institución.

Referencias

- [1] Microcontrolador. (2018, 31 de septiembre). *Wikipedia*.
Obtenido de <https://es.wikipedia.org/w/index.php?title=Microcontrolador&oldid=109640416>.
- [2] ¿Qué es Arduino? (2018, 29 de septiembre). Arduino.cl:
Obtenido de <http://arduino.cl/que-es-arduino/>
- [3] ¿Qué es un sistema rfid? (2018,19 de noviembre). *Wikipedia*.
Obtenido de <https://es.wikipedia.org/wiki/RFID>
- [4] Elementos de un sistema rfid. (2018 ,19 de noviembre)
Obtenido de https://www.kimaldi.com/blog/rfid/elementos_del_sistema_rfid/
- [5] Bases de Datos. (2018 ,19 de noviembre)
Obtenido de https://techlandia.com/ejemplos-programas-administracion-bases-datos-lista_126285/
- [6] Módulo RFID RC-522. (2018, 19 de noviembre)
Obtenido de http://hnaylampmechatronics.com/blog/22_Tutorial-Lector-RFID-RC522.html
- [7] Sistemas de identificación. (2018, 19 de noviembre)
Obtenido de <http://www.dipolerfid.es/es/blog/Tipos-Sistemas-RFID>
- [8] Ethernet shield. (2018, 19 de noviembre)
Obtenido de <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>
- [9] tags. (2018, 19 de noviembre)
Obtenido de <https://es.wikipedia.org/wiki/RFID>
- [10] Principios básicos. (2018, 20 de noviembre)
Obtenido de <https://www.fqingenieria.com/es/conocimiento/principios-basicos-de-la-tecnologia-rfid-39>
- [11] RFID. (2018, 26 de noviembre). *Wikipedia, La enciclopedia libre*
Obtenido de <https://es.wikipedia.org/w/index.php?title=RFID&oldid=112283735>.
- [12] Programas para Bases de Datos. (2018, 20 de enero)
Obtenido de <https://docs.kde.org/trunk5/es/calligra/kexi/database-creation-software.html>
- [13] jQuery. (2013, 4 de mayo). desarrollo web.
Obtenido de: <https://desarrolloweb.com/articulos/introduccion-jquery.html>
- [14] Tags rfid pasivos. (2017). desarrollo web.
Obtenido de: <http://www.comercialpuelo.cl/site/trovan/rfid/>

- [15] Comunicación SPI. (2018, 1 de octubre). Wikipedia, La enciclopedia libre
Obtenido de: https://es.wikipedia.org/wiki/Serial_Peripheral_Interface
- [16] EXCEL. (2018, 1 de octubre). exceltotal
Obtenido de: <https://exceltotal.com/que-es-excel/>
- [17] Manejo de datos en EXCEL. (2018, 1 de octubre).
Obtenido de: <https://es.scribd.com/doc/18588044/EXCEL-para-manejo-de-datos#scribd>

ANEXOS

ANEXO 1: ARDUINO UNO DATASHEET

ANEXO 2: ETHERNET SHIELD DATASHEET

ANEXO 3: RC-522 DATASHEET

**ANEXO 4: AGREGAR NUEVOS ESTUDIANTES Y ASOCIAR
IDENTIFICACIÓN DE TARJETA**