



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Medida Remota de la Temperatura de un Interferómetro de Sagnac dentro de una Red LAN

TESIS

Para obtener el grado de
Ingeniero en Redes



PRESENTA

Abraham Sierra Calderón

DIRECTOR DE TESIS

Ing. Francisco Méndez Martínez

ASESORES

Dr. Homero Toral Cruz

Dr. Freddy Ignacio Chan Puc

Dr. Víctor Manuel Sánchez Huerta

Dr. Jaime Silverio Ortegón Aguilar





UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Trabajo de Tesis elaborado bajo supervisión del Comité de asesoría y aprobada
como requisito parcial para obtener el grado de:

INGENIERO EN REDES

Comité de Trabajo de Tesis



Director:

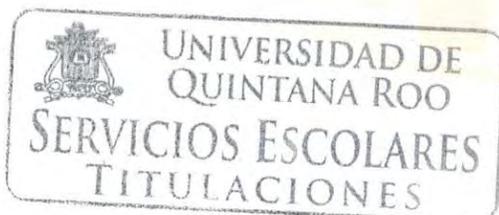
Ing. Francisco Méndez Martínez

Asesor:

Dr. Homero Toral Cruz

Asesor:

Dr. Freddy Ignacio Chan Puc



Chetumal Quintana Roo, México, Noviembre de 2015

Dedicatoria

Después de unos años de haber concluido con la carrera profesional de Ingeniería en Redes y posteriormente la elaboración de la tesis para obtener mi grado de estudio como Ingeniero en Redes, le dedico este trabajo primero que nada a Dios por haberme permitido concluir mis estudios con buena salud y con el tiempo estimado.

También quiero dedicarle de todo corazón a mi mamá María del Socorro Calderón y Villanueva, de igual manera a mi papá Pedro Enrique Sierra Pacheco ya que fueron las personas que más me apoyaron de forma económica y moralmente, en todos los estudios desde el kínder hasta ahora lo que hoy es la Universidad, siempre estuvieron alentándome y apoyándome en lo que más necesité, quiero agradecerles ya que esto es la mejor herencia que pudiera haber obtenido y de igual manera este es un regalo hacia ellos ya que demuestra todo el sacrificio que realizaron a lo largo de estos 24 años.

A mi hermana Alejandra Sierra Calderón que siempre ha estado a mi lado apoyándome en todo lo que necesité y alentándome a seguir este camino de superación personal.

Al Dr. Manuel May Alarcón, al Dr. Francisco Méndez Martínez, al Dr. Homero Toral Cruz y a todos los profesores involucrados en este trabajo de tesis, ya que ofrecieron un gran tiempo a la realización de este trabajo desde la elaboración del anteproyecto hasta el asesoramiento de la redacción de esta tesis.

A todos los profesores involucrados en la realización de las enseñanzas de diversas materias llevadas a lo largo de la carrera por su tiempo y asesoramiento en cualquier horario que hayan tenido disponible.

A todas las personas y amigos en general que estuvieron a lo largo de estos años, en los buenos y malos momentos, en especial a mi novia Wendy Abigail Chalé Montalvo, por sus relajos, desvelos y apoyo mutuo para poder llevar esta etapa universitaria de la mejor manera.

Agradecimientos

Quiero agradecer primero que nada a la Universidad de Quintana Roo por aceptarme en la carrera y por todo su apoyo en los dispositivos a utilizar, en las becas que ofrece para poder experimentar otras ramas fuera de la ciudad y en todo lo que se necesitó para poder aprender de una mejor manera dicha carrera, de igual manera le dedico este trabajo de tesis a todos los maestros involucrados en mi educación universitaria dentro del plantel, ya que sin ellos esta meta no se hubiera podido lograr.

A la Universidad Autónoma del Carmen por aceptarme para estudiar un semestre en sus instalaciones en el área de Fibras Ópticas junto con sus laboratorios y dispositivos, de igual manera a los profesores el Dr. Francisco Méndez Martínez y el Dr. Manuel May Alarcón por todo su apoyo en las becas, en la formación del estudio universitario, como amigos mientras me encontraba radicando en Ciudad Del Carmen y por permitirme y apoyarme a realizar este trabajo de tesis en dicha universidad.

A mi familia y a mis amigos que me estuvieron apoyando a lo largo de todos estos años, ya que si no hubiera sido por ellos no me encontraría en este momento realizando la finalización de mi carrera universitaria.

TABLA DE CONTENIDO

TABLA DE CONTENIDO	iii
ÍNDICE DE FIGURAS	vi
ÍNDICE DE TABLAS	vii
RESUMEN	viii
CAPÍTULO 1. MEDIDA REMOTA DE LA TEMPERATURA DE UN INTERFERÓMETRO DE SAGNAC DENTRO DE UNA RED LAN	1
1.1 Antecedentes.....	1
1.2 Identificación del problema	2
1.3 Propuesta de solución al problema	3
1.4 Justificación	5
1.5 Objetivos	5
1.5.1 Objetivo general.....	5
1.5.2 Objetivos específicos	5
1.6 Metodología	6
1.7 Productos esperados	7
CAPÍTULO 2. BASES TEÓRICAS	8
2.1 Introducción.....	8
2.2 Conceptos de electrónica.....	8
2.2.1 Temperatura	9
2.2.2 Sensores de temperatura	9
2.2.3 Arduino.....	12
2.3 Conceptos de óptica	14
2.3.1 Arreglo óptico	14
2.3.2 Interferómetro de Sagnac de fibra óptica	16
2.4 Conceptos de redes	17
2.4.1 Medida remota	18
2.4.2 Red LAN	19
2.5 Resumen.....	25
CAPÍTULO 3. ARREGLO EXPERIMENTAL	26
3.1 Introducción.....	26
3.1 ¿Por qué se hace?.....	26
3.2 ¿Cómo se hace?.....	28
3.2.1 Verificación de la conexión LAN	29
3.2.2 Dispositivos a utilizar	30

3.2.3 Software a utilizar	31
3.2.4 Creación y conexión del circuito	31
3.2.5 Creación del software	32
3.2.5.1 Java	33
3.2.5.2 HTML, PHP Y MySQL	34
3.2.6 Instalación y configuración de los servidores	34
3.2.6.1 Servidor de acceso remoto	35
3.2.6.2 Servidor de base de datos	35
3.2.6.3 Servidor web	36
3.2.6.4 Servidor FTP	36
3.2.7 Realizar pruebas de verificación	37
3.3 Resumen	37
<i>CAPÍTULO 4. RESULTADOS Y ANÁLISIS.....</i>	38
4.1 Introducción.....	38
4.2 Resultados	38
4.2.1 Resultado arreglo óptico	38
4.2.2 Resultado circuito del sensor de temperatura DS18B20 con el Arduino MEGA 2560	39
4.2.3 Resultado código de temperatura para el Arduino	42
4.2.4 Resultado conexión y configuración de un servidor para un acceso remoto	44
4.2.5 Resultado creación de una aplicación con el lenguaje de programación en Java	47
4.2.6 Resultado conexión con un servidor FTP para la transferencia de archivos	48
4.2.7 Resultado creación de un sistema de búsqueda profunda para los datos de temperatura guardada en una base de datos.....	51
4.3 Resumen.....	55
<i>CAPÍTULO 5. CONCLUSIONES</i>	56
<i>BIBLIOGRAFÍA</i>	58
<i>ANEXOS.....</i>	60
Código en Arduino para un sensor	60
Código en Arduino para dos sensores	61
Código en la plataforma Java para un sensor	62
Apartado de Java Class	62
Apartado del JFrame	63
Código en la plataforma Java para dos sensores	71
Apartado Java Class	71
Apartado del JFrame	72
Código de la página web para un sensor	81
Conexión a la base de datos de MySQL.....	81
Inicio de sesión.....	82
Control de verificación.....	82
Validación de contraseña.....	83
Index.....	84

Cambio de contraseña	92
Información del contacto.....	95
Motor de búsqueda por fecha	97
Motor de búsqueda por hora	102
Motor de búsqueda por temperatura	107
Respaldo en formato PDF por medio de fecha	112
Respaldo en formato PDF datos por hora	114
Respaldo en formato PDF por temperatura.....	116
Hoja de estilos para la página web.....	117
Código de la página web para dos sensores	121
Home	122
Busqueda por fecha	128
Busqueda por hora.....	133
Busqueda por la temperatura del primer sensor	137
Busqueda por la temperatura del segundo sensor.....	142
Respaldo en formato PDF con datos en general	146
Respaldo en formato PDF por fecha.....	147
Respaldo del formato en PDF por hora	148
Respaldo en formato PDF por la temperatura del primer sensor	149
Respaldo en formato PDF por la temperatura del segundo sensor	150
Datasheet sensor de temperatura DS18B20.....	151

ÍNDICE DE FIGURAS

Figura 1. Diagrama esquemático de la Red LAN dentro de la Universidad Autónoma del Carmen	4
Figura 2. Diagrama esquemático para la medición de la temperatura de un interferómetro de Sagnac	4
Figura 3. Sensor de temperatura DS18B20	11
Figura 4. Placa microcontroladora Arduino MEGA 2560 con 54 pines digitales y 16 analógicas	13
Figura 5. Diagrama esquemático para la medición de la temperatura de un Interferómetro de Sagnac dentro de una Red LAN	15
Figura 6. Interferómetro de Sagnac	17
Figura 7. Red LAN (Local Area Network)	20
Figura 8. El modelo OSI	21
Figura 9. Arquitectura cliente-servidor	22
Figura 10. Arquitectura punto a punto	22
Figura 11. Topología de estrella	23
Figura 12. Topología de anillo	24
Figura 13. Topología de bus	24
Figura 14. Arreglo del servidor remoto	27
Figura 15. Arreglo de la conexión remota dentro de la Universidad Autónoma del Carmen	28
Figura 16. Diagrama para la realización del trabajo de tesis	29
Figura 17. Circuito eléctrico para los sensores de temperatura DS18B20	32
Figura 18. Resultado del arreglo óptico utilizado en la tesis	39
Figura 19. Conexión del Arduino con el sensor DS18B20	40
Figura 20. Plantilla del circuito en PCB	40
Figura 21. Shield de temperatura para Arduino, creado de forma casera	41
Figura 22. Shield de temperatura para Arduino, creado de forma casera	42
Figura 23. Datos del sensor DS18B20 programados en Arduino	44
Figura 24. Diagrama de la tesis dentro de una Red LAN	45
Figura 25. Conexión cliente – servidor	45
Figura 26. Conexión servidor – cliente	46
Figura 27. Conexión remota	46
Figura 28. Monitoreo de la temperatura	47
Figura 29. Respaldo de la captura y monitoreo de los datos de temperatura en Excel	48
Figura 30. Prendiendo el servidor FTP por medio del software XAMPP	49
Figura 31. Configurando el servidor FTP con un usuario y una contraseña	49
Figura 32. Conectando el servidor FTP desde el cliente hacia el servidor	50
Figura 33. Transferencia de archivos desde el servidor hacia el cliente	51
Figura 34. Página principal del sistema de búsquedas	52
Figura 35. Motor de búsqueda	53
Figura 36. Cambio de contraseña	53
Figura 37. Información del contacto	54
Figura 38. Ejemplo del motor de búsqueda	54

ÍNDICE DE TABLAS

<i>Tabla 1. Dispositivos de medición de la temperatura.....</i>	<i>11</i>
---	-----------

RESUMEN

Los sistemas de comunicación mediante fibras ópticas son unos de los sistemas más atractivos debido a que permiten realizar la transmisión de información a altas velocidades. Estos sistemas ofrecen numerosas ventajas relacionadas principalmente con las siguientes características: confiabilidad, aislamiento eléctrico, baja pérdida de transmisión, seguridad, tamaño reducido, bajo peso, elevado ancho de banda, eliminación de cortos. Sin embargo, es de suma importancia el monitoreo y control de algunas variables que tienen influencia en el funcionamiento óptimo de ciertos dispositivos que forman un sistema de comunicación óptico. Una de las variables físicas que pueden tener una importante influencia en la transmisión de información en un sistema óptico es la temperatura. El interferómetro de Sagnac es muy utilizado en diversas configuraciones de sistemas ópticos para eliminar el ruido producido por la amplificación de emisión espontánea de dichos sistemas. Por tal motivo, el monitoreo y captura constante de la temperatura en este dispositivo es de gran relevancia para verificar su correcto funcionamiento.

Por otro lado, una de las tendencias actuales es la automatización de procesos, con el objetivo de hacer uso más eficiente del tiempo, en este trabajo de tesis se propone una plataforma de comunicación remota para realizar el monitoreo de la temperatura de un interferómetro de Sagnac que se encuentra dentro de un arreglo óptico. La plataforma de comunicación propuesta, se basa en la arquitectura cliente-servidor, permite el acceso remoto en un entorno de Red LAN y está desarrollada mediante la combinación de los siguientes elementos de hardware y software: cliente, servidor de acceso remoto, servidor de base de datos, servidor web, servidor FTP, sensor de temperatura, PCB, Arduino MEGA 2560 e interfaz de usuario web.

CAPÍTULO 1. MEDIDA REMOTA DE LA TEMPERATURA DE UN INTERFERÓMETRO DE SAGNAC DENTRO DE UNA RED LAN

1.1 Antecedentes

Desde que se empezaron a realizar las comunicaciones entre equipos de cómputo, las primeras conexiones en red fueron las Redes de Área Local (*LAN, Local Area Network*), después se fue expandiendo a las Redes de Área Metropolitana (*MAN, Metropolitan Area Network*) y Redes de Área Amplia (*WAN, Wide Area Network*) por medio del muy conocido *WWW (World Wide Web)* [1], desde ahí se han creado servidores web (Red, Internet, *WWW*) (Hobbs, 1999) [2], diversos servidores para comunicaciones, y de igual manera servidores para la medición de diferentes sensores aplicados por ejemplo en estaciones meteorológicas a las cuales se puede tener acceso remoto ya sea en una Red LAN o en una WAN como el Internet [3], [4].

Las conexiones remotas dentro de una Red LAN, MAN o WAN han tenido un gran auge hoy en día, debido a que no es necesario que una persona se traslade físicamente a un lugar específico para realizar alguna tarea o actividad, como participar en una conferencia, caracterizar datos o llenar formularios. En trabajos de investigación y desarrollo tecnológico, es importante tener una conexión remota para detectar parámetros físicos, tales como: temperatura, presión, tensión y aceleración.

Se puede definir como conexión remota, a la operación de conectarse a una red o computadora desde un punto lejano dentro de una misma red o ajeno a esa red, usando la conectividad de una Red LAN o WAN donde uno de ellos permite acceder al otro como si se estuviese trabajando directamente enfrente de este sin necesidad de su presencia física, al utilizar un software de acceso remoto [5].

Dentro de un edificio, escuela, empresa, Cibercafé, o centro de juegos electrónicos, las computadoras o video juegos necesitan estar conectados entre sí para poder compartir datos, voz y video, a este tipo de conexión local se le llama Red LAN que

está compuesta por un conjunto de computadoras que se conectan entre sí en un área geográficamente limitada a 5 km de distancia, como un edificio, una fábrica o un campo universitario. En una LAN, cualquier computadora puede comunicarse con todas las demás y compartir recursos [6].

En la presente tesis se desarrollará una plataforma para tener acceso de manera remota, a las mediciones de temperatura de un interferómetro de Sagnac Dentro de una Red LAN. En donde se desarrollará en el estado de Campeche en Cd. Del Carmen, dentro de las instalaciones de la Universidad Autónoma del Carmen (UNACAR) específicamente en el edificio de Ingenierías, el lugar donde se medirá la temperatura de un interferómetro de Sagnac es en el laboratorio de Óptica, situado en el segundo piso, y el lugar donde se realizará la medición de dicha temperatura por medio de un acceso remoto se encuentra en un cubículo ubicado en el primer piso. Esta medición solo se podrá realizar dentro de una Red LAN, que en este caso sería en el área de Ingeniería de la Universidad Autónoma del Carmen, posterior mente se podría hacer la medición de forma WAN, en cualquier área del mundo, siempre y cuando se cuente con acceso a Internet.

1.2 Identificación del problema

El interferómetro de Sagnac es muy utilizado en diversas configuraciones de sistemas ópticos, entre sus aplicaciones más comunes puede ser usado: para eliminar el ruido producido por la Emisión Espontanea Amplificada (*ASE Amplified Spontaneous Emission*), como multiplexor [7], [8], como espejo no lineal [7], [9], como sensor de fibras ya sea con alta o baja birrefringencia [7], [10], [11], entre otras aplicaciones más. Por otro lado, la temperatura es una de las variables físicas que más se mide, por ejemplo, la temperatura tiene efectos significativos en la reflexión y en la transmisión del interferómetro de Sagnac. Por tal motivo, el monitoreo y captura constante de los cambios de temperatura en este dispositivo es de gran relevancia para verificar su correcto funcionamiento.

Para poder medir y monitorear la temperatura de un dispositivo físico, el usuario tiene que estar presente en el lugar que se encuentre instalado para poder verificar

y tomar la lectura de la medición, lo cual resulta impráctico debido al tiempo invertido en ir y venir al lugar donde se encuentre ubicado dicho experimento que es sometido a cambios de temperatura. A lo expresado anteriormente se presenta un grave problema de inversión de tiempo de desplazamiento para ir y tomar medidas, además de que se pierde dinero si la distancia de desplazamiento es muy grande, es decir, si un experimento se encuentra en un estado y la persona encargada de recabar dicha información se encuentra en otro. Es por esta misma razón que la Universidad Autónoma del Carmen tiene este problema, ya que no se puede invertir todo el tiempo posible estando en el lugar del experimento, ya que necesitan realizar otras actividades de igual importancia, por ahora es difícil estarse dividiendo en actividades y en tiempo para poderlo aprovecharlo de la mejor manera.

1.3 Propuesta de solución al problema

Para resolver el problema de que un usuario tenga que ir a medir y capturar la variable física hasta donde se encuentre desarrollando la medición de la temperatura en un interferómetro de Sagnac, se creará una plataforma de comunicación remota que sirva de enlace entre el sensor de temperatura del interferómetro y el usuario. La plataforma propuesta estará formada por medio de un servidor web o escritorio remoto, un servidor FTP para la transferencia de los datos, y una base de datos, en la figura 1 se puede observar el diagrama esquemático de la Red LAN que se encuentra dentro de la universidad, en dicha red se puede observar dos equipos de cómputo, uno en cada lugar de forma separada, el cliente se encuentra en el piso B1 y el servidor en el piso B2, ambos conectados entre sí. Para emular diversos escenarios con diferentes temperaturas, se utilizará una placa de Peltier, la cual estará controlada por medio de una fuente de voltaje y la medición de la temperatura se realizará con el sensor DS18B20, tal y como se muestra en la figura 2 que muestra el diagrama de la tesis dentro del laboratorio, en donde se podrá encontrar la conexión de diferentes dispositivos para la medición y monitoreo de la temperatura.

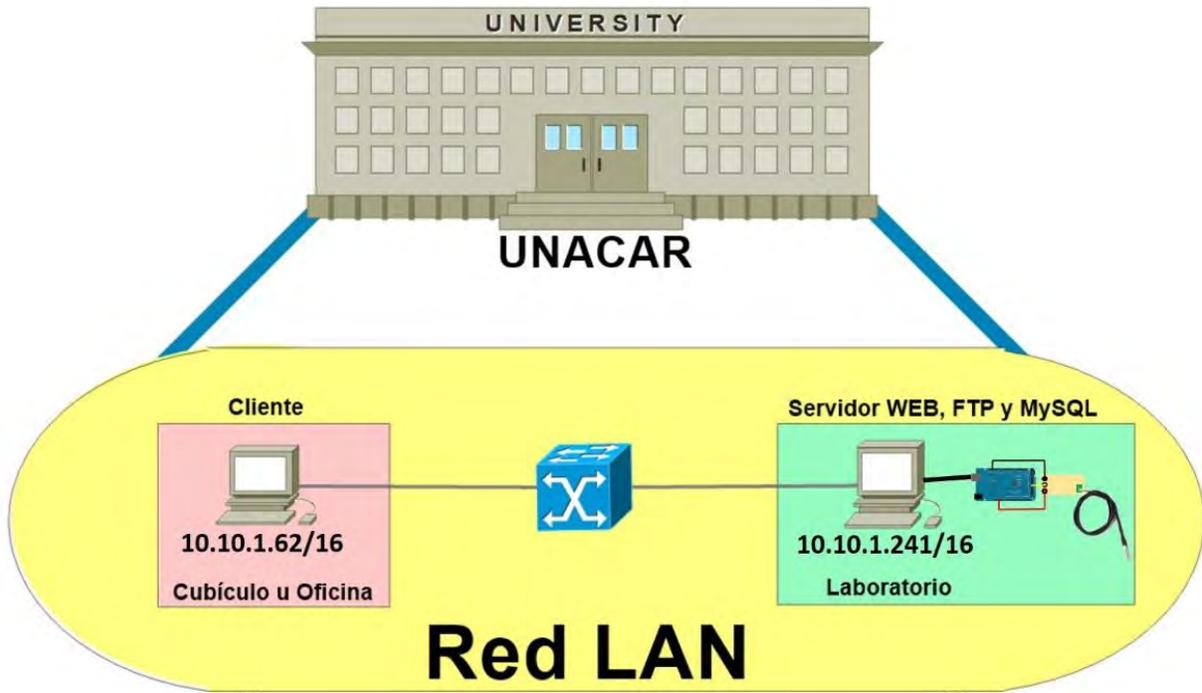


Figura 1. Diagrama esquemático de la Red LAN dentro de la Universidad Autónoma del Carmen

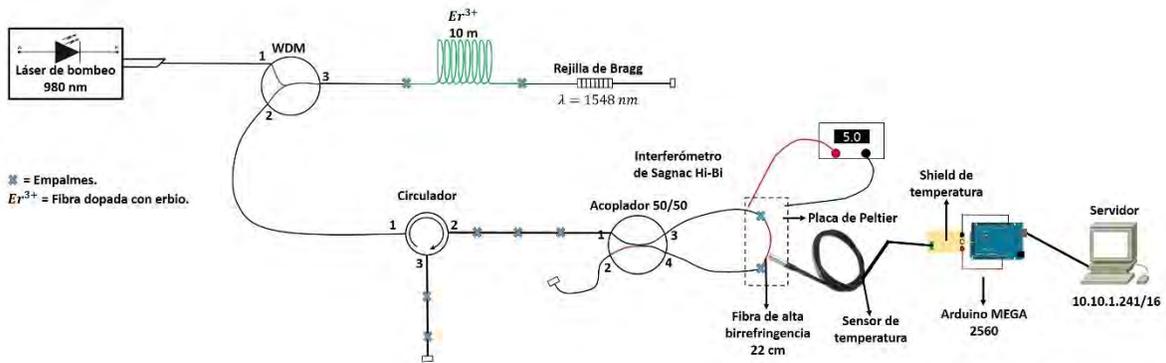


Figura 2. Diagrama esquemático para la medición de la temperatura de un interferómetro de Sagnac

1.4 Justificación

Como se mencionó anteriormente, es importante medir la temperatura en un interferómetro de Sagnac, porque este tiene importante impacto en la potencia reflejada y la potencia transmitida. Debido a que en la actualidad, una de las tendencias es la automatización de procesos, con el objetivo de hacer un uso más eficiente del tiempo, en este trabajo de tesis se propone una plataforma de comunicación remota para realizar el monitoreo de la temperatura de un interferómetro de Sagnac.

Por medio de dicha conexión remota el usuario va a poder realizar la medición y monitoreo desde su cubículo u oficina y de igual manera podrá estar realizando otros tipos de trabajos mientras se hace la captura de temperatura sin necesidad de estar presente en el laboratorio o donde se encuentre el experimento.

1.5 Objetivos

1.5.1 Objetivo general

Medir y monitorear la temperatura a la cual se encuentra sometido un interferómetro de Sagnac para que sea enviada de manera remota dentro de una Red LAN por medio de un dispositivo Arduino Mega 2560.

1.5.2 Objetivos específicos

Para poder cumplir el objetivo general de una manera más efectiva y con el menor tiempo posible, se crearon puntos, estos puntos son objetivos específicos que al realizarlos al pie de la letra de deberá cumplir el objetivo general.

- Solicitar los permisos necesarios para incorporar un equipo de cómputo a la Red LAN.
- Verificar que la conexión LAN esté en funcionamiento.
- Levantar los servidores requeridos para el envío de datos.
- Programar en una plataforma web con conexión a Java y así mismo con conexión al Hardware Arduino Mega 2560, conectarla a una base de datos y levantarlo en el servidor.
- Conectar el circuito del sensor de temperatura y corroborar que su funcionamiento sea adecuado.
- Hacer pruebas para verificar que el servidor y el sensor estén en funcionamiento.

1.6 Metodología

En el subtema anterior “objetivos específicos”, se plantearon algunos puntos, algunos de estos puntos se encuentran de una forma general. En este subtema, “la metodología”, se despliegan aún más los objetivos específicos con diferentes puntos para que después de realizarlo se puedan completar los objetivos específicos y de igual manera el objetivo general.

- Verificar que haya conexión entre el cliente y el servidor.
- Obtener los dispositivos necesarios para la programación en Arduino.
- Realizar el programa en Arduino Mega 2560 para poner en marcha el sensor de temperatura DS18B20.
- Realizar un programa en la plataforma Java para hacer que capture los datos de temperatura en una base de datos y que los guarde en formato Excel y PDF.
- Utilizar el Software Team Viewer en el servidor y en el cliente para realizar el escritorio remoto e instalar el Software Xampp para el servidor FTP, web y MySQL.
- Instalar el Software FileZilla Client en el cliente para poder conectarse al servidor y se pueda hacer la transferencia de datos.

- Realizar un programa en la plataforma HTML y PHP para realizar un sistema de búsqueda avanzada en el servidor MySQL.
- Hacer pruebas de captura de temperatura y del respaldo de información entre el cliente y el servidor.
- Analizar todos los resultados obtenidos del sistema propuesto.

1.7 Productos esperados

Finalmente, una vez que se hayan cumplidos los objetivos específicos, se tendrán que ver los resultados de este, a continuación se despliegan algunos puntos de los resultados que se tendrán que tener al poder realizar y cumplir con el objetivo general de este tema de tesis.

- Implementación del sistema propuesto.
- Configuración y aplicación del sistema que detecta y envía datos de temperatura.
- Documentación del código de las aplicaciones ya planteadas.
- Difusión de resultados.
- Redacción y defensa de la tesis.

CAPÍTULO 2. BASES TEÓRICAS

2.1 Introducción

Hoy en día existen infinidad de términos muy sofisticados para todas las ramas de estudio, gracias a estos términos cualquier usuario con conocimientos muy sólidos o básicos se puede dar una idea de lo que se está estudiando en algún tema de investigación, es por eso que en este capítulo se presenta un marco teórico, para que cualquier lector que no contenga noción o hasta el que tenga noción con el tema pueda entender el trabajo de tesis ya planteado en el capítulo uno.

En este capítulo dos se van a plantear tres tipos de conceptos, los conceptos de electrónica, óptica y redes. Para finalizar con un resumen del capítulo.

2.2 Conceptos de electrónica

La electrónica es un tema fundamental en este trabajo de tesis, ya que sin ella no se podría hacer ninguna medición y mucho menos poderla visualizar en alguna pantalla ya que cualquier dispositivo a utilizar en este proyecto funciona con arreglos electrónicos. A continuación se va a poder observar en los subcapítulos 2.2.1, 2.2.2 y 2.2.3 algunos conceptos sobre la electrónica que se va a estar utilizando en este tema de tesis, para que el lector se pueda dar una idea de lo que se va a realizar al mencionar los conceptos en los capítulos posteriores, el significado de los conceptos fueron sacados de fuentes de información como libros, libros electrónicos, diccionarios y libros de investigación.

2.2.1 Temperatura

La temperatura es un término utilizado día a día en actividades de todos los aspectos, primordialmente para mantener con vida a cualquier ser, también desde que uno se levanta hasta cuándo se encuentra dormido, se puede encontrar en la ventilación del aire acondicionado y natural, al realizar una comida, al refrigerar los alimentos y en actividades muy diversas que podamos imaginarnos. Pero ¿Qué es la temperatura? Todos los seres vivos podemos sentirla pero no verla, sabemos que la temperatura va cambiando con respecto al tiempo.

Es por eso que se puede definir la temperatura como la propiedad de un cuerpo que determina la dirección del flujo de calor cuando se pone en contacto con otro cuerpo, el cual pasa del cuerpo de temperatura más alta al de temperatura más baja. Viene definida por el principio cero de termodinámica. La temperatura se percibe por la sensación de caliente y frío y se mide con una escala basada en alguna propiedad adecuada [12].

Por ejemplo se puede medir la temperatura de un cuerpo humano con un termómetro, al hacer contacto estos dos cuerpos la temperatura más alta que es la del cuerpo humano, esta pasa a la temperatura más baja que es la del termómetro y es así como se puede saber cuál es la temperatura del cuerpo humano.

2.2.2 Sensores de temperatura

Se definió anteriormente qué es la temperatura, la temperatura no se puede ver pero si se puede sentir, conforme a la sensación se puede verificar un grado de temperatura. ¿Cómo se puede verificar este grado de temperatura?, por medio de un sensor, el cual puede ser un termómetro o un dispositivo electrónico con diferentes propiedades, el termómetro contiene un rango de dígitos visibles para poder observar la temperatura y así poder realizar la medición cuando el calor es transferido a este cuerpo.

Para poder observar el grado de temperatura de algún objeto sólido se tiene que utilizar un sensor, el cual es un dispositivo el cual proporciona una salida utilizable en respuesta a la medida especificada. La salida es definida como una cantidad eléctrica y la medida es una cantidad física, propiedad o condición de la cual es medida. Los sensores proporcionan información de ciertas variables que se rodean para poder procesarlas y así generar ordenes o activar procesos [13].

En la actualidad existen muchas formas de medir la temperatura con todo tipo de sensores de diversas naturalezas. En la tabla 1 se podrá observar una muestra de la gran variedad de dispositivos capaces de medir la temperatura. El sensor de temperatura que se utilizará en esta tesis es el sensor digital DS18B20 sumergible de tipo RTD como se puede observar en la figura 3. El sensor DS18B20 es un dispositivo que se comunica de forma digital. Cuenta con tres terminales, el primero es de voltaje, el segundo de tierra y el tercero de datos. Este sensor trabaja con una potencia desde 3V hasta 5.5V y su medición de la temperatura contiene un rango de -55°C hasta los 125°C . La característica más importante de este sensor de temperatura es que trabaja por medio de un solo cable, es decir con el protocolo OneWire, este protocolo permite el envío y la recepción de datos por un solo cable. Para más información sobre este sensor de temperatura, puede consultar en el área de anexos el datasheet del sensor.

DISPOSITIVOS DE MEDICION DE TEMPERATURA	
Eléctricos	
Termocuplas	
Termorresistencias	
Termistores	
Diodos	
Sensores de silicio con efecto resistivo	
Mecánicos	
Sistemas de dilatación	
Termómetros de vidrio con líquidos	

Termómetros bimetálicos	
Radiación térmica	
Pirómetros de radiación	
	-Total (banda ancha)
	-Óptico
	-Pasabanda
	-Relación
Termómetros infrarrojos	
Varios	
Indicadores de color	
	-Lápices
	-Pinturas
Sondas neumáticas	
Sensores ultrasónicos	
Indicadores pirométricos	
Termómetros acústicos	
Cristales líquidos	
Sensores fluidos	
Indicadores de iluminación (Termografía)	

Tabla 1. Dispositivos de medición de la temperatura



Figura 3. Sensor de temperatura DS18B20

2.2.3 Arduino

Conforme va pasando el tiempo, se van creando nuevas tecnologías y el estilo de vida de todas las personas se va acoplado conforme a estas nuevas tecnologías, gracias a esto existe mayor trabajo pero menor tiempo para realizarlo, algunas tecnologías nuevas que se están utilizando hoy en día son los pagos por Internet y hasta por celular, ahora se pueden encontrar que los datos y el estilo de vida de las personas se reducen en un solo dispositivo como fotos, videos, datos personales y hasta en qué lugar se encuentra, es por eso que se van creando nuevos proyectos innovadores para cada situación y resolución de problemas para las personas. Por medio de este hardware llamado Arduino se han estado creando diversos proyectos con diversos sensores y arreglos electrónicos, como la realización de una casa autónoma, alguna medición con algún sensor de temperatura, ultrasónico, envió de alarmas a teléfonos celulares, desde una computadora que se encuentre conectada al Arduino hasta traspasar los datos a nivel mundial por medio del Internet, se preguntarán ¿Qué es ese hardware Arduino y cómo funciona?

Es por eso que a continuación se define como Arduino a una tarjeta microcontroladora diseñada para crear procesos y proyectos usando componentes electrónicos, este hardware consiste en una tarjeta diseñada con código abierto, su lenguaje de programación compila y se ejecuta con un gestor de arranque en el microcontrolador [14].

También se puede decir que el Arduino es una pequeña computadora en donde se puede programar para procesar entradas y salidas entre los dispositivos y componentes externos que se le conecten [14].

Existen varios modelos de placas Arduino, al igual que modulos y Shield, algunas placas que se pueden encontrar en el mercado son: Arduino UNO, Arduino PRO, Arduino MEGA, Arduino ZERO, Arduino DUE, Arduino YÚN, Arduino GEMMA y Aruino LILYPAD [15].

La placa Arduino que se está utilizando para esta tesis es la placa Arduino MEGA 2560 como se puede observar en la figura 4. El Arduino MEGA 2560 es una tarjeta

microcontroladora que cuenta con 54 pines digitales de entrada o salida las cuales 15 pueden ser usadas para la salida del PWM (Pulse Width Modulation), 16 pines para entradas analógicas, una conexión USB, un power Jack y un botón para el reseteo del código. Este hardware Arduino opera con un voltaje de 5V, contiene una memoria de 256 kb de los cuales 8 kb se usan para el software de arranque y para poder programar esta tarjeta microcontroladora se tiene que utilizar el software de Arduino (IDE). De las características ya descritas, en este tema de tesis se utilizará el puerto USB serial, el power jack para la alimentación del voltaje, un pin para alimentar al sensor, un pin para tierra y un pin digital que se encontrará funcionando como OneWire.

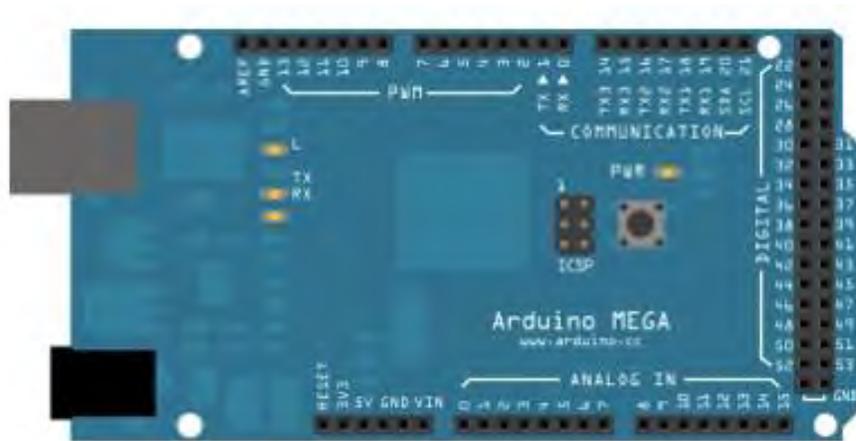


Figura 4. Placa microcontroladora Arduino MEGA 2560 con 54 pines digitales y 16 analógicas

2.3 Conceptos de óptica

La óptica es un concepto que se utiliza a diario sin que se den cuenta, ya que es la rama de la física que analiza las características y las propiedades de la luz. Entonces, si hay luz, hay óptica. La luz se puede observar en cualquier momento ya sea como luz coherente (láser), luz incoherente (led, luz del sol, luz de la luna, vela), luz no visible y luz visible. Gracias a la óptica es que en la actualidad existen diversos dispositivos que se utilizan día a día, como los televisores, controles remotos, los microondas y los rayos x. Los conceptos que se presentarán a continuación son conceptos ópticos que se estarán utilizando a lo largo de esta tesis ya que son la base de este proyecto.

2.3.1 Arreglo óptico

Desde que empiezan a salir los primeros rayos del sol hasta que anochece se utilizan arreglos ópticos o visiones ópticas, desde que podamos observar cualquier objeto y podamos visualizar sus colores, en ese momento se está utilizando la óptica, la óptica se puede encontrar en sus medios naturales como son los rayos del sol, de la luna, en una vela prendida o un candelabro de petróleo, también se puede encontrar en un entorno artificial como es el televisor, los celulares, los focos incandescentes o ahorradores, en los leds, etc.

La óptica funciona desde un solo rayo o fotón, para algunos experimentos o casos particulares se necesita guiar ese rayo o fotón por medio de algún medio o una guía, es por eso que se crearon los arreglos ópticos. Donde un arreglo óptico es la creación y conexión de distintos dispositivos electrónicos y ópticos los cuales envían y reciben luz o fotones, la longitud de la onda puede estar dentro del rango visible y no visible. Un ejemplo de un arreglo óptico es la transmisión infrarroja, ya que un dispositivo óptico envía luz infrarroja y otro dispositivo óptico recibe esa misma luz infrarroja, como los celulares y los controles de los televisores.

El arreglo óptico que se utilizará para este tema de tesis, es un arreglo que fue creado anteriormente en la tesis del Ingeniero en Redes José Carlos Novelo

Rodríguez que tiene como título “Caracterización de los Componentes Ópticos en un Sistema que Atenúa Ruido ASE por Medio de un Interferómetro de Sagnac que Funciona como Filtro Pasa Banda” por medio de la Universidad de Quintana Roo y la Universidad Autónoma del Carmen como se puede observar en la figura 5, este arreglo óptico está formado por un diodo láser de 980 nm y con diversos tipos de fibras ópticas. El objetivo principal de este arreglo óptico es atenuar el ruido ASE, como está compuesto por una fibra de alta birrefringencia, esta es sensible a los cambios de temperatura, es por esta misma razón que se hace el monitoreo de la misma para poder ir observando los cambios que realiza el ruido ASE con respecto a la temperatura y de esa misma manera poder calcular la mejor manera de atenuar dicho ruido.

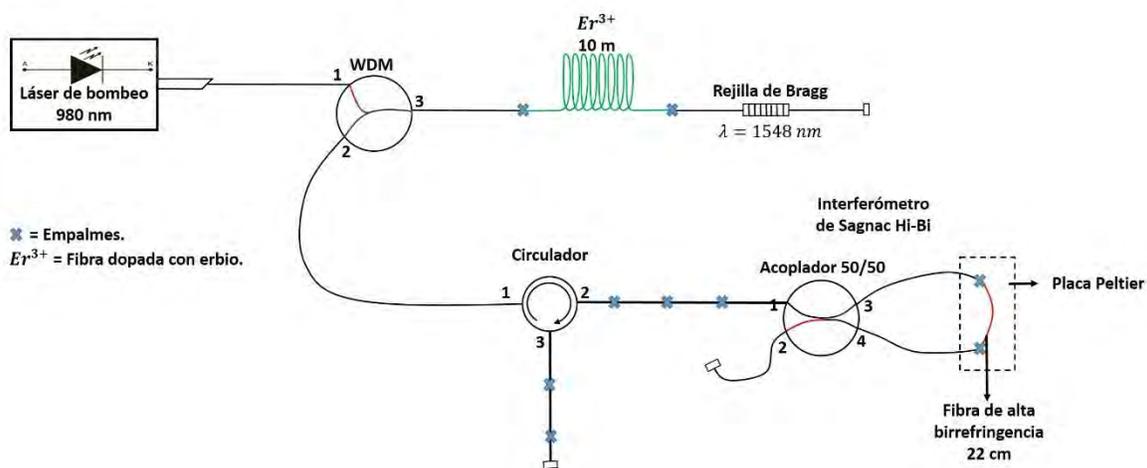


Figura 5. Diagrama esquemático para la medición de la temperatura de un Interferómetro de Sagnac dentro de una Red LAN

2.3.2 Interferómetro de Sagnac de fibra óptica

Al haber utilizado el arreglo óptico antes mencionado de la tesis del Ingeniero en Redes José Carlos Rodríguez Novelo, se obtuvo que aparte de reducir el ruido ASE y aumentar la potencia, el ruido se puede reducir aún más y así poder seguir aumentando la potencia, la clave o el arreglo específico para realizar este proceso es el interferómetro de Sagnac.

El interferómetro de Sagnac de fibra óptica también llamado espejo de lazo óptico no lineal está compuesto por un acoplador 50/50, 90/10, 99/1 o 75/25 de cuatro puertos y de fibra de alta o baja birrefringencia, el acoplador hace que la potencia de la luz de entrada uno se divida en dos señales en las salidas tres y cuatro con potencias iguales o diferentes de acuerdo a las características del acoplador. Una vez que se ha dividido el haz, cada una de las potencias viaja dentro del lazo de fibra, en direcciones contrarias uno en sentido horario y otro en sentido anti horario, la interferencia se da cuando el par de pulsos regresa hacia el acoplador; Es allí donde la interferencia se presenta, sin embargo para que la interferencia dependa de la potencia se debe introducir una asimetría en el esquema, por lo tanto, estos haces viajan a lo largo del lazo de fibra pero al pasar por la fibra de alta o baja birrefringencia este haz de luz se desfasa, para entrar nuevamente al acoplador para recombinarse y producir un haz de salida que se conoce como señal transmitida la cual se encuentra en la terminal dos como se observa en la figura 6. El interferómetro de Sagnac a utilizar en esta tesis consta de un acoplador 50/50, el cual se encarga de dividir la potencia de la luz en un 50% para la terminal tres y un 50% para la terminal cuatro. El interferómetro de Sagnac también cuenta con un pedazo de fibra de alta birrefringencia entre la terminal tres y cuatro, esta fibra se encontrará sometida a cambios de temperatura por medio de una placa Peltier que se le agregara posteriormente al interferómetro de Sagnac, a la placa de Peltier le va a estar llegando cierta cantidad de voltaje que convertirá en temperatura, y así es como se va agregando cierta cantidad de temperatura a la fibra de alta birrefringencia como se puede observar en la figura 6, al interferómetro de Sagnac se le puede medir la potencia transmitida y la potencia reflejada.

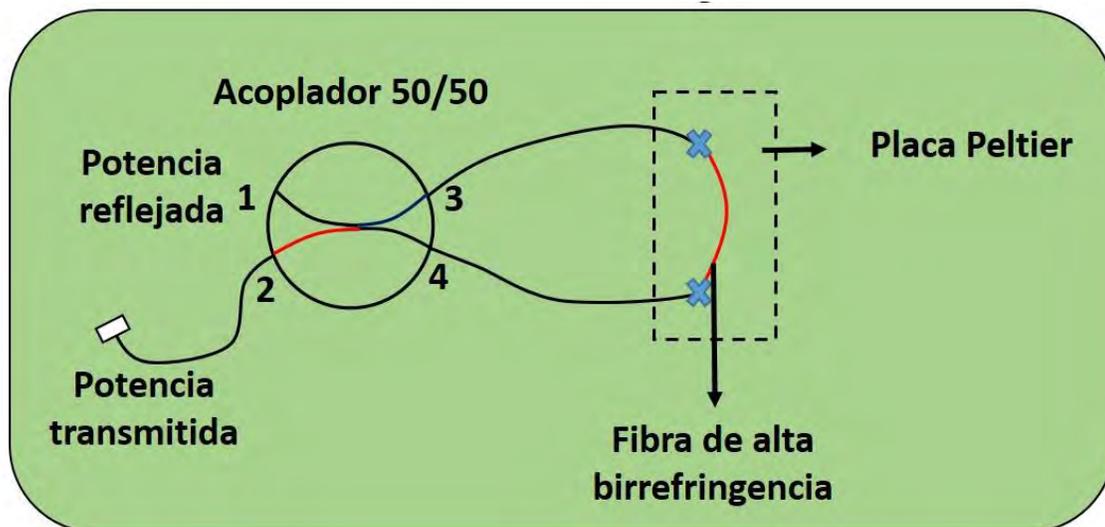


Figura 6. Interferómetro de Sagnac

2.4 Conceptos de redes

Las redes de las telecomunicaciones son bastante importantes hoy en día, ya que la comunicación se utiliza a diario y gracias a las telecomunicaciones el mundo va teniendo cada día nuevos avances tecnológicos, a continuación se verán algunos conceptos sobre redes de telecomunicaciones que se estarán utilizando en esta tesis, ya que es el medio que se encargará para poder hacer las mediciones de temperatura de manera remota y así poder enviarlo a un equipo de cómputo para que el usuario pueda medir y analizar los resultados.

2.4.1 Medida remota

Las medidas y los números siempre han existido desde que el ser humano se encuentra en la tierra, son palabras que se escuchan y se utilizan a diario, desde que suena el despertador hasta la hora que duerme una persona, sin embargo, se debe tener claramente qué significa una medida, una medida o medición es una unidad de una expresión numérica la cual se emplea para medir longitudes, áreas, volúmenes o cualquier otra variable. [16]

La palabra remota anteriormente se utilizaba muy poco, en la actualidad se puede escuchar más veces o puede ser familiar utilizar la palabra remota como en un control remoto ya sea para un juguete o para la televisión, esta palabra de igual manera se puede encontrar en términos electrónicos, en computadoras y celulares, entonces el significado de la palabra remota es algo lejano en el tiempo o en el espacio [17].

Al realizar la unión de estos dos términos concluimos con la palabra medición remota, la cual es muy importante en esta tesis, ya que con esto se va a poder hacer la medición de una temperatura con un Arduino y un sensor de temperatura DS18B20 a un arreglo óptico.

Con la unión de estos dos términos se puede concluir que la medición remota es la unidad que se emplea para medir alguna variable específica en una cierta distancia lejana o en el espacio. Por ejemplo: se pueden hacer mediciones meteorológicas por medio de mediciones remotas ya que el usuario no tiene que estar en el lugar físicamente para poder hacer la medición.

2.4.2 Red LAN

Desde el tiempo de la prehistoria han habido varias formas de comunicación, como las señales de humo, después las claves morse, la carta, el fax, el teléfono, el teléfono a ido actualizándose hasta convertirse en un teléfono digital, a partir de ahí se ha convertido en un punto final de la comunicación como las computadoras, tabletas y laptops, ya que el medio de comunicación a nivel mundial que hoy en día existe es la red del Internet, existen diversos tipos de comunicaciones, pero para poder llegar a estas comunicaciones tuvo que haber una gran evolución de las tecnologías en comunicaciones, un medio muy grande para las comunicaciones es el Internet, el Internet contiene infraestructuras, topologías y estándares que deben cumplirse, antes del Internet hubo varios tipos de redes en comunicaciones, si se regresan a los años 80 en ese año fue la aparición de las redes LAN la cual dio un salto enorme en la evolución de las telecomunicaciones [18] hasta llegar hoy en día lo que es el Internet. La Red LAN es un conjunto de computadoras que se encuentran conectadas entre sí, en una cierta área geográficamente limitada como puede observarse en la figura 7 [19], las computadoras dentro de una Red LAN pueden enviarse recursos entre sí como voz, datos y videos. Una Red LAN puede ser una oficina pequeña, una fábrica, una casa, un cibercafé o un campo Universitario. Dentro de una Red LAN se pueden encontrar recursos compartidos como son las impresoras, mensajería instantánea y servidores, la transmisión de estos recursos pueden llegar a alcanzar velocidades hasta de 100 Mbps [19].

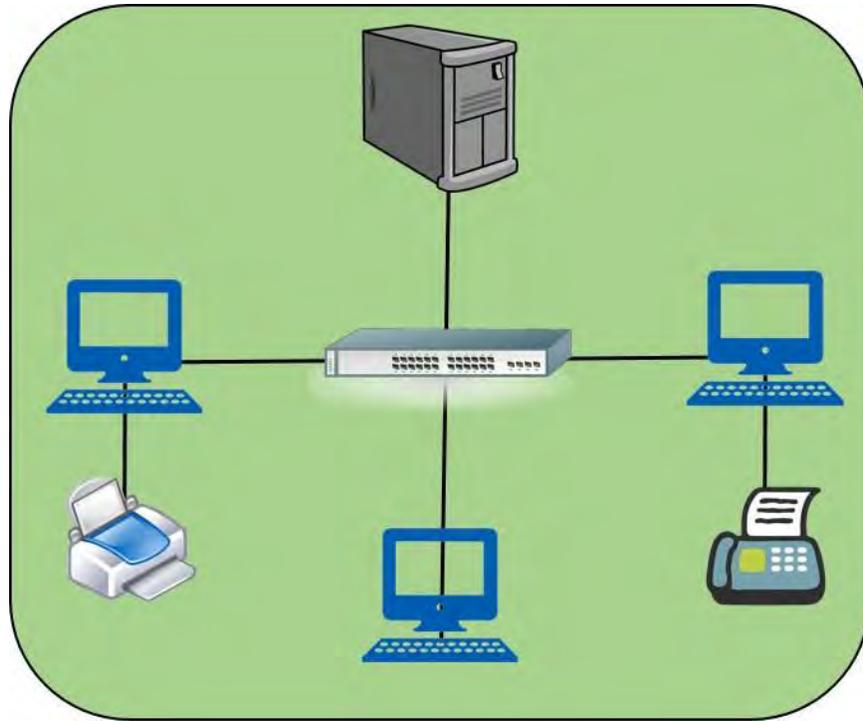


Figura 7. Red LAN (Local Area Network)

Al crearse distintos equipos de cómputo por diferentes fabricantes, un problema mayor era crear redes con los diferentes equipos ya que no había un estándar que concuerde entre estos, es por eso que el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) creó un estándar llamado el 802 [19], el cual se encarga de poder comunicar computadoras entre sí, sin importarle de cual fabricante sea y así poder expandir las conexiones en redes LAN. Este estándar fue desarrollado simultáneamente en cooperación con el modelo OSI (Open System Interconnection), el cual es un modelo básico de referencia que lo que hace es resolver los problemas de las comunicaciones de datos y las redes informáticas dividiéndolas en siete niveles como se puede observar en la figura 8. Para que haya comunicación dentro de una Red LAN, se utilizan tres capas establecidas en el modelo OSI, uno (Física), dos (Enlace de datos) y tres (Red) [20].

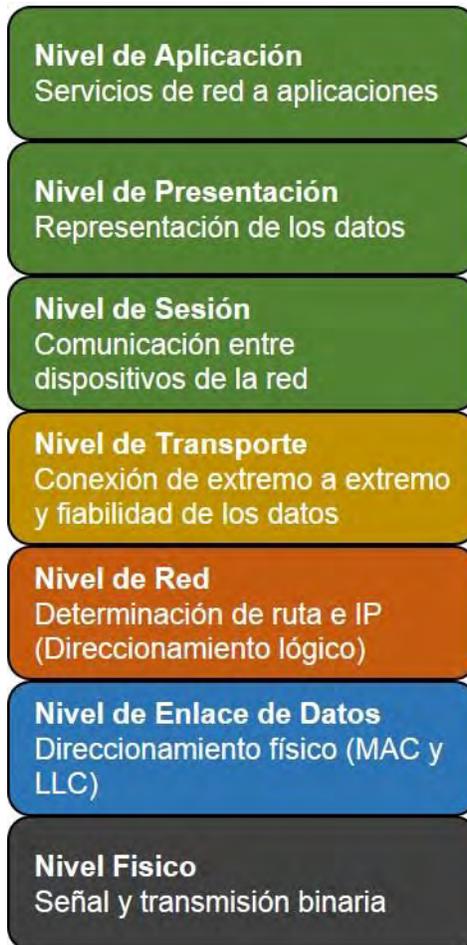


Figura 8. El modelo OSI

Las redes LAN utilizan un tipo de procesamiento llamado procesamiento distribuido, quien es el encargado de conectar distintos equipos de cómputo, en donde los datos pueden ser ejecutados o procesados en varias máquinas de la red. Dentro del procesamiento distribuido se pueden encontrar dos tipos de arquitectura, conexión cliente-servidor y conexión punto a punto [19]. La conexión cliente-servidor es donde el servidor se pone como dedicado, es decir, que contendrá todas las aplicaciones usadas en la red y el cliente para poder usar estas aplicaciones tendrá que dirigirse al servidor como se puede observar en la figura 9 y la conexión punto a punto es donde los equipos de cómputo dentro de la red se configuran de manera que los servidores no sean dedicados, ya que todas las computadoras podrán compartir recursos o aplicaciones entre sí como se puede observar en la figura 10.

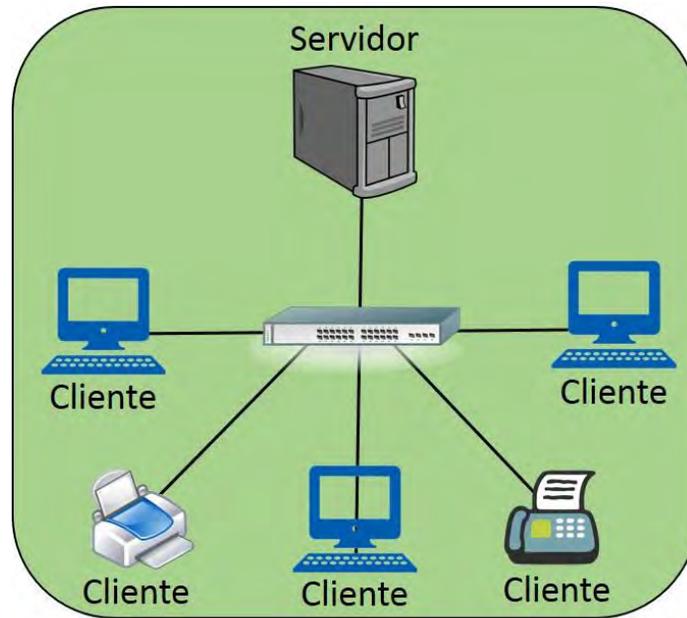


Figura 9. Arquitectura cliente-servidor

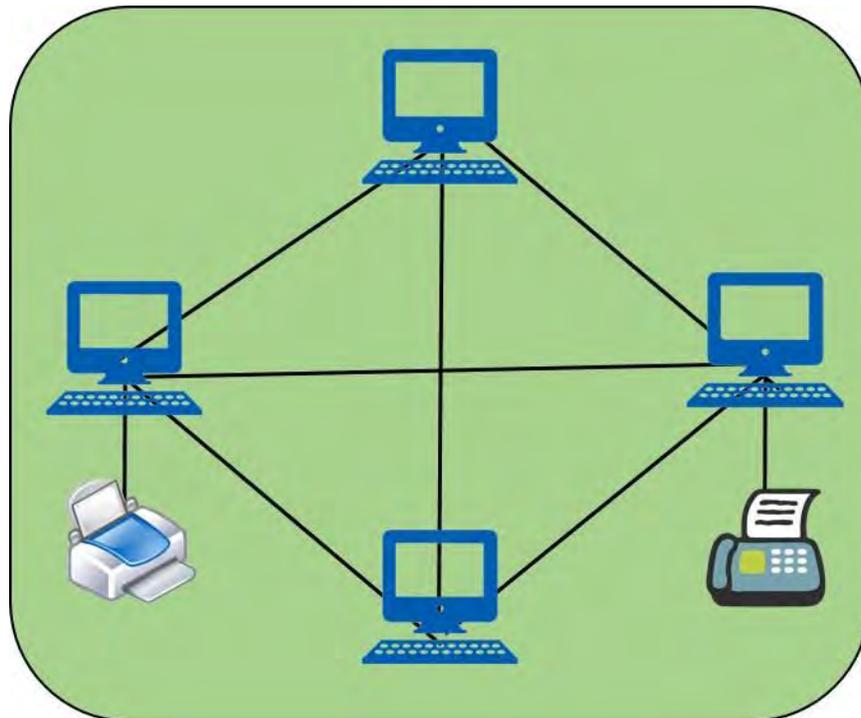


Figura 10. Arquitectura punto a punto

Dentro de las redes LAN se pueden observar sus estándares que se caracterizan por el tipo de topología, las topologías más usadas dentro de las redes LAN son las de estrella que es donde todos los equipos de cómputo están conectados entre sí por medio de un Switch, Hub o Router, las de anillo, en donde todos equipos de cómputo están conectados entre sí, sin la necesidad de estar conectados por medio de un Switch, Hub o Router, su conexión es punto a punto y la topología de bus que es donde todos los nodos van conectados a un solo circuito común o bus, la información que se envía en el bus pasa por las computadoras que sean necesarias hasta llegar a su destino como se pueden observar en las figuras 11, 12 y 13 [19].

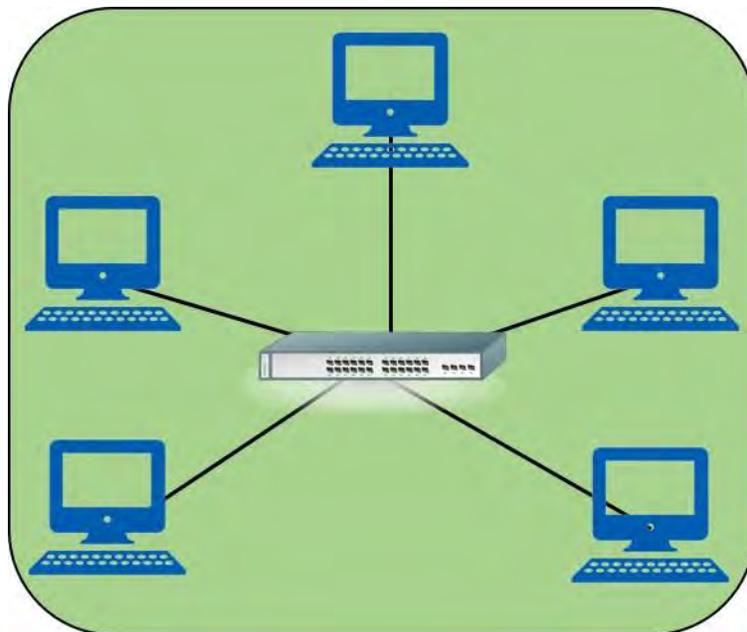


Figura 11. Topología de estrella

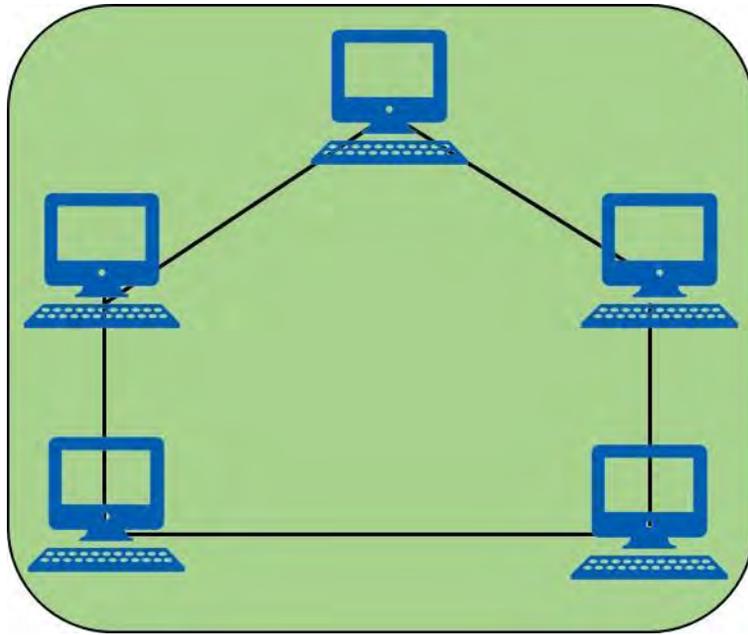


Figura 12. Topología de anillo

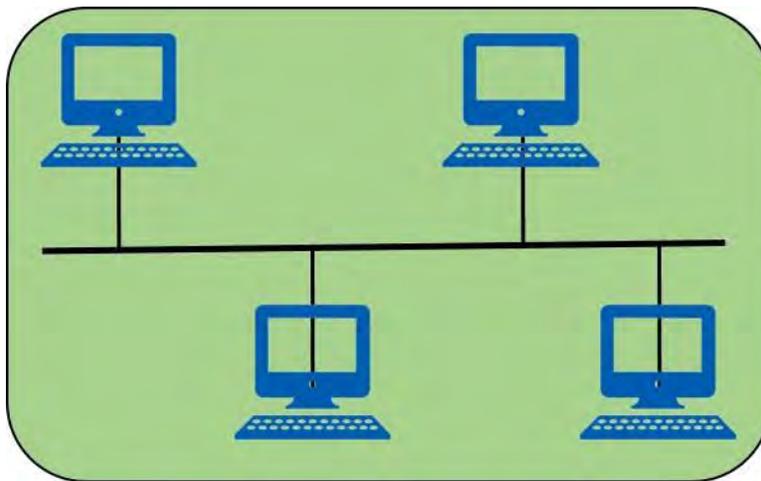


Figura 13. Topología de bus

2.5 Resumen

En el presente capítulo se abordaron los conceptos básicos de electrónica, óptica y redes. Dichos conceptos fueron claves para entender la problemática tratada en este trabajo de tesis y proporcionaron las herramientas necesarias en la solución propuesta. Dentro de los conceptos de electrónica se definió el proceso de detección y medición de la temperatura mediante un sensor, y la manipulación de esta variable física por medio de un dispositivo llamado Arduino para la correcta visualización por parte de un usuario. Por otro lado, se definió un arreglo óptico sensible a los cambios de temperatura, capaz de atenuar el ruido ASE mediante el interferómetro de Sagnac. Finalmente, se describió un sistema de comunicación LAN basado en la arquitectura cliente-servidor, capaz de transportar la variable física temperatura, medida sobre el arreglo óptico mencionado.

CAPÍTULO 3. ARREGLO EXPERIMENTAL

3.1 Introducción

En este capítulo se presentará la importancia del porque se realiza y como se realiza esta tesis, ya que juega un papel muy importante en la experimentación. Se podrá observar las herramientas, dispositivos y software a utilizar para que posteriormente se empiecen a realizar los objetivo específicos ya planteados en el capítulo uno y así poder llegar al objetivo general de esta tesis. Se podrá observar la combinación de la electrónica, las configuraciones de dispositivos informáticos y la utilización de los lenguajes de programación para que posteriormente se puedan mostrar los resultados y así poder haber cumplido el objetivo general. Los puntos que se verán a continuación se deben de entender a la perfección para que se pueda resolver la gran problemática que se planteó en el capítulo uno.

3.1 ¿Por qué se hace?

Para poder hacer la medición de la temperatura de un interferómetro de Sagnac, es necesario un sensor de temperatura, el sensor de temperatura que se va a utilizar es el DS18B20, el cual va a estar midiendo la temperatura de la fibra de alta birrefringencia ya que es muy sensible a los cambios de temperatura y por esta misma razón al estar dentro de un interferómetro de Sagnac se podrá eliminar el ruido ASE de la señal de entrada, esta fibra se encuentra ubicada en el arreglo óptico antes mencionado en el capítulo dos, en la figura 5 y para pasar esos datos de temperatura a una forma visible para el usuario, se tendrá que estar conectado a un Arduino Mega 2560 y enlazado por medio de un puerto serial a un servidor como se puede ver en la figura 14 en donde se muestra el arreglo de la conexión del sensor de temperatura al hardware Arduino junto con su servidor.

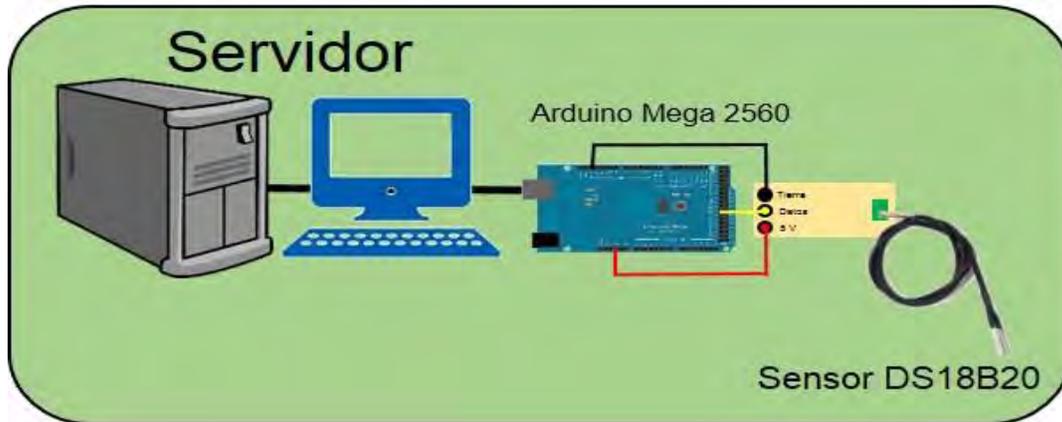


Figura 14. Arreglo del servidor remoto

Posteriormente para disminuir el mayor tiempo perdido al trasladarse desde un lugar origen, hasta donde se encuentre el experimento y hacer las mediciones de la temperatura físicamente. En la figura15 se podrá observar el arreglo del sensor con el Arduino, ahora estarán conectados en una Red LAN para poder hacer la medición de la temperatura pasando los datos de temperatura por la placa de Peltier a la fibra de alta birrefringencia y después al sensor DS18B20 que estará conectado al Arduino y al servidor para que el usuario pueda visualizar estos datos de temperatura, después estos datos serán enviados de forma remota a la computadora donde el usuario estará haciendo dicha medición dentro de la Red LAN, pero ahora por medio de un acceso remoto. Es decir, que no habrá necesidad de trasladarse físicamente al lugar del experimento, si no que ahora se podrá hacer desde una oficina o en cualquier computadora conectada a esa misma Red LAN. Gracias a esto se va a poder ahorrar un mayor tiempo de trabajo y se va a poder aprovechar ese mismo tiempo para poder realizar otras actividades, ahora los usuarios podrán ser más efectivos a la hora de realizar su trabajo.

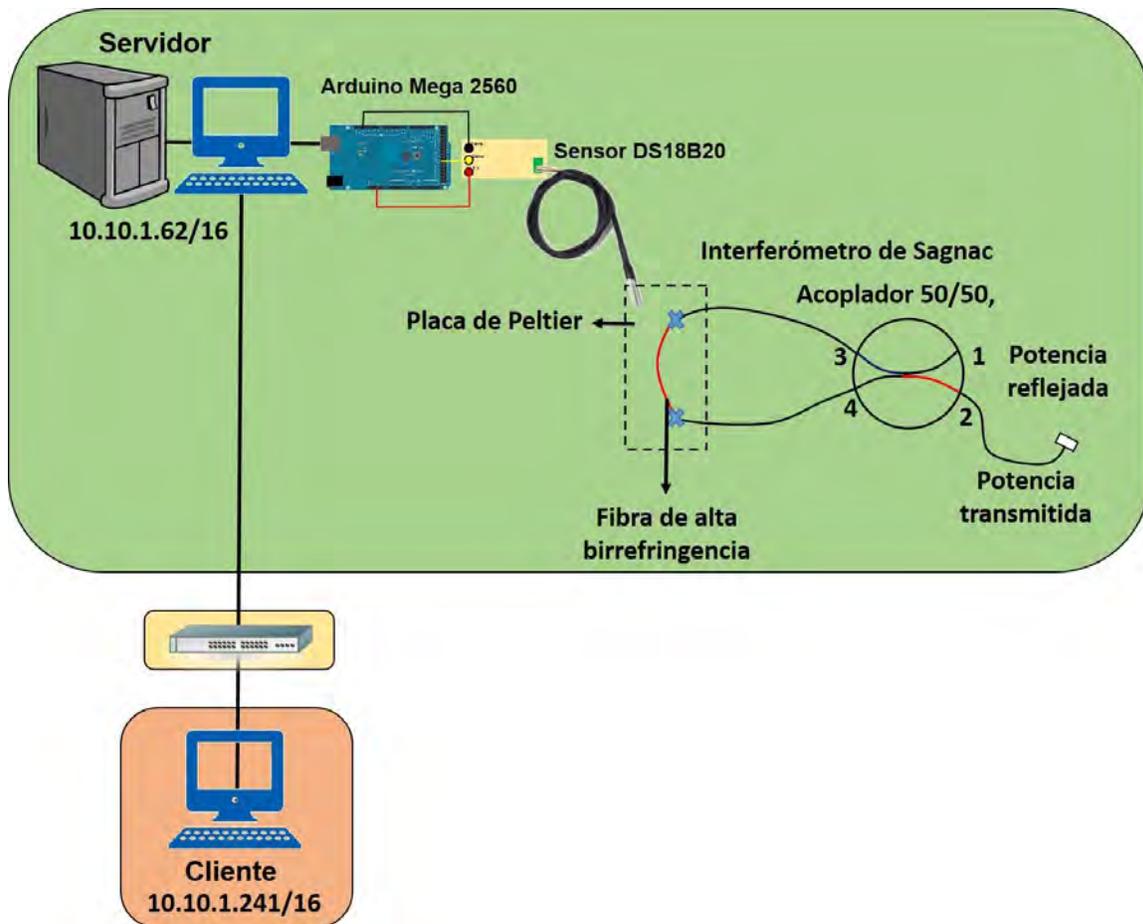


Figura 15. Arreglo de la conexión remota dentro de la Universidad Autónoma del Carmen

3.2 ¿Cómo se hace?

Se logró entender cuál es la problemática y el objetivo, ahora habrá que saber qué es lo que se tendrá que hacer para poder llegar a ese objetivo y de esa manera poder darle una solución a la problemática ya planteada, a continuación se va a presentar en la figura16 un diagrama de cómo se va a realizar la solución al problema de esta tesis titulada “**Medida Remota de la Temperatura de un Interferómetro de Sagnac dentro de una Red LAN**”.

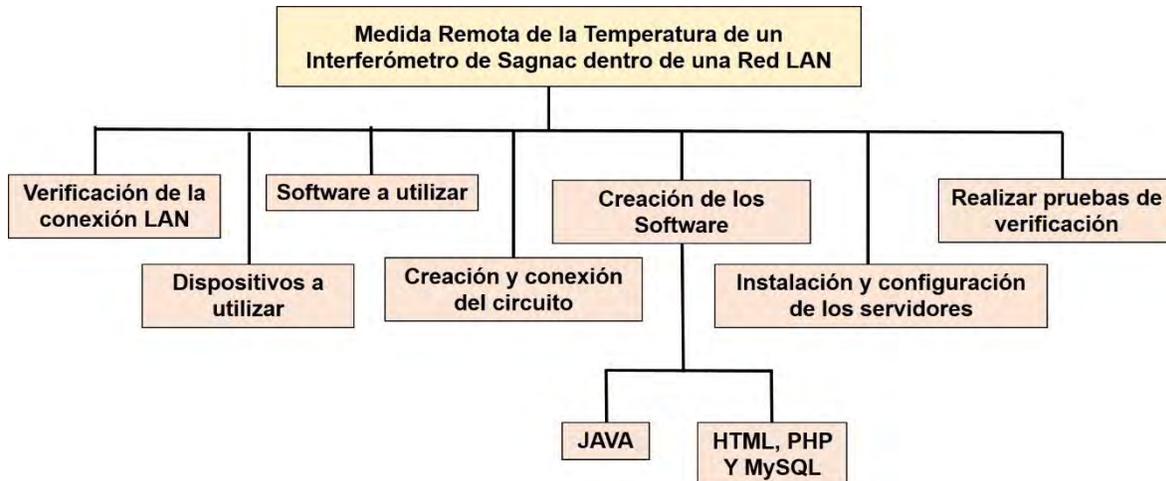


Figura 16. Diagrama para la realización del trabajo de tesis

3.2.1 Verificación de la conexión LAN

Dentro del plantel de ingeniería de la Universidad Autónoma del Carmen en el estado de Campeche, se estará realizando esta tesis, la universidad consta de varios departamentos, como la dirección, escolar, de redes y telecomunicaciones, etc.

Para poder agregar un equipo de cómputo a la Red LAN de la universidad, no se puede nada más llegar, agregarle una dirección IP(Internet Protocol) al equipo y conectarlo, lo que se va a realizar va a ser un oficio en donde dirá que se le dará de alta a un equipo de cómputo dentro de la Red LAN de la universidad, dicho equipo de cómputo se va a situar dentro del laboratorio de óptica, ya que ahí se va a encontrar el arreglo óptico que contendrá el interferómetro de Sagnac, posteriormente el departamento de redes y telecomunicaciones aceptara dicha petición y se encargará de conectar el equipo de cómputo dentro de la Red LAN de la Universidad y luego le pondrá una dirección IP que se encuentre dentro del rango de dicha red.

La Red LAN a la que se necesitará agregar el equipo de cómputo contará con una dirección IP de clase A, ya que estará dentro del rango de la dirección IP 10.10.1.0/16.

Una vez que se haya dado de alta al equipo de cómputo se tendrá que saber si verdaderamente existe conexión a la Red LAN, esto se va a realizar por el método llamado PING el equipo agregado tendrá que hacerle ping al equipo que funcionará como cliente por medio del comando ping 10.10.1.62 (dirección IP del cliente), los resultados deben ser exitosos ya que ningún paquete se debe perder, si el ping es exitoso entonces habrá comunicación en la Red LAN.

Otra manera de verificar que el nuevo equipo de cómputo este dentro de la Red LAN de la universidad es por medio del comando ipconfig /all, nos mostrará una serie de parámetros, verificamos que la dirección IP este dentro del rango de la dirección IP de la Red LAN junto con su máscara de red.

3.2.2 Dispositivos a utilizar

Para poder realizar esta tesis, se va a tener que utilizar algunos dispositivos para tener una conexión cliente-servidor y para poder realizar la medición de la temperatura, los dispositivos a utilizar son:

- Dos equipos de cómputo, uno funcionará como cliente y otro como servidor.
- Arreglo óptico con un interferómetro de Sagnac.
- Arduino Mega 2560.
- Dos sensores de temperatura DS18B20.
- Un Shield de temperatura para conectar los sensores DS18B20.

3.2.3 Software a utilizar

Los software que se utilizarán para esta tesis nos ayudara a levantar servicios como el control de acceso remoto, servidor FTP, servidor web, servidor de base de datos, creación de un sistema en Java y sobrescribir código para el dispositivo Arduino, los software que se utilizarán son:

- Team Viewer.
- Xampp.
- NetBeans IDE 8.0.2.
- Sublime Text2.
- Arduino 1.6.5.

3.2.4 Creación y conexión del circuito

Para poder crear el circuito que va a hacer la medición de la temperatura, primero se va a tener que conectar el Arduino al equipo de cómputo que va a funcionar como servidor por medio de un cable serial, se procederá a conectar el sensor de temperatura al Arduino pero antes se creará un circuito el cual funcionará como una Shield de temperatura, la placa o circuito que se creará contendrá dos resistencias de $5k\Omega$ a $6k\Omega$, estas resistencias estarán conectadas a las salidas de corriente el cual va a ir conectado al Arduino que cuenta con 5V y a un pin digital que contendrá el Arduino, aparte habrá una línea exclusivamente para tierra, la cual se conectará de igual manera al Arduino, dentro de la misma placa irán conectados seis terminal blocks en cada uno podrá ir conectado un sensor de temperatura, esta tesis va a trabajar para dos sensores, pero este proyecto tendrá la posibilidad a expandirse a más sensores. En los terminal blocks irán conectados las líneas de los sensores los cuales son de alimentación, de datos y tierra, la conexión del circuito electrónico se puede visualizar en la figura 17. El sensor de temperatura que estará conectado al Arduino y seguidamente al servidor, va a estar conectado al arreglo óptico del interferómetro de Sagnac.

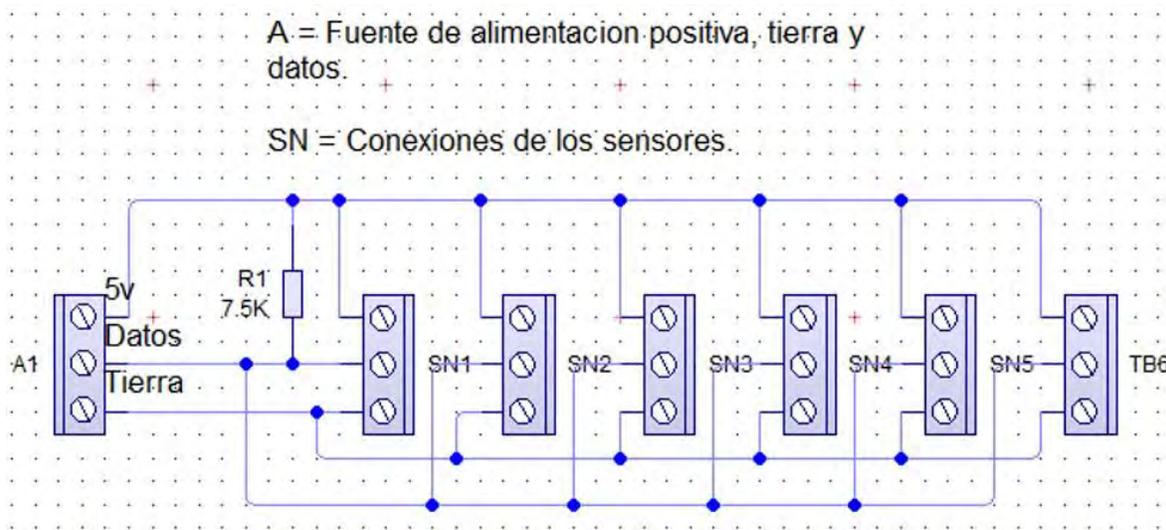


Figura 17. Circuito eléctrico para los sensores de temperatura DS18B20

3.2.5 Creación del software

El sensor junto con el Arduino van a poder hacer tomas de los datos de temperatura que vaya arrojando el interferómetro de Sagnac, para que el usuario pueda visualizar estos datos y posteriormente graficarlos para ver su comportamiento, se tendrá que crear dos aplicaciones, una es para realizar la toma de datos, su visualización y su respaldo, y la otra aplicación es para poder visualizar de una mejor manera los datos por medio de motores de búsqueda y también poder observar su comportamiento por medio de una gráfica. Para la primera aplicación que es la toma, visualización y el respaldo de los datos, se tendrá que usar el software de Android IDE 1.6.5 para poder programar la placa Arduino junto con el sensor DS18B20 y así poder hacer la toma de datos, para que esta toma de datos se vea más estético y posteriormente poder hacer su respaldo, se utilizará el software NetBeans IDE 8.0.2 para poder programar una aplicación con el lenguaje en Java. Para la segunda aplicación que es la visualización y traficación de los datos por medio de un motor de búsqueda, se va a utilizar el software sublime text2 para programar una página web que estará hecho por medio de etiquetas HTML5, lenguaje de programación PHP y JavaScript, para poder visualizar la página web en un navegador web como lo es Internet Explorer, Google Chrome, Firefox, etc.

3.2.5.1 Java

La aplicación que se va a realizar en la plataforma Java por medio del software NetBeans se podrán observar los datos arrojados que el sensor de temperatura va a ir tomando, se van a poder visualizar en °C para uno o hasta para dos sensores, no solo se podrán visualizar los datos de la temperatura, sino también la hora y la fecha en que fueron tomados esos datos, habrá posibilidad de aumentar el número de sensores a visualizar, se podrán guardar los datos tomados en un formato .XML para poder tener un respaldo y posteriormente poder graficarlos, de igual manera los datos a visualizar se podrán ir guardando automáticamente en una base de datos para posteriormente tenerlos cuando los necesitemos, se entiende que la temperatura juega un papel importante al momento de estar utilizando el interferómetro de Sagnac, es por eso que se necesita saber con qué nivel de temperatura se está trabajando y saber hasta que temperatura no se puede sobrepasar. Se le agregará una advertencia la cual aparecerá cuando la temperatura sobrepase sus límites y así poder bajar e ir controlando dicha temperatura.

3.2.5.2 HTML, PHP Y MySQL

Una vez terminada la aplicación en Java se pueden visualizar los datos de la temperatura junto con la hora y fecha en que fueron tomadas, posteriormente se podrá hacer un respaldo de dichos datos en un formato .XML. mientras se hace la toma de datos en la plataforma Java se van a ir guardando automáticamente esos datos en una base de datos, ahora se va a realizar una plataforma web en donde el usuario va a poder acceder por medio de un usuario y una contraseña, ya una vez dentro de la aplicación web el usuario va a poder observar todos los datos que se tomaron por medio de la aplicación en Java, de igual manera se va a poder ver una gráfica de esos mismos datos por medio de la temperatura y sus horario, para poder utilizar todos estos datos de una mejor manera se van a agregar búsquedas específicas, por fecha, por hora y por temperatura, de esta manera el usuario podrá observar los datos más detenidamente y solo los que les interese, también podrá visualizar las gráficas de los datos seleccionados, todos estos datos podrán guardarse en un documento en PDF para poder tenerlo a la mano cuando se desee, las gráficas podrán guardarse en formato PDF, JPG y GIF, para una mejor seguridad el usuario podrá cambiar la contraseña de su cuenta las veces que desee.

3.2.6 Instalación y configuración de los servidores

Es muy importante levantar adecuadamente y configurar los servidores a utilizar, ya que de ellos depende que la solución a la problemática sea exitosa, y con esto se cumple el objetivo deseado, los servidores a levantar y configurar son los siguientes:

3.2.6.1 Servidor de acceso remoto

Con este servidor se va a poder realizar la toma de los datos de la temperatura remotamente, no es necesario encontrarse físicamente en el lugar del experimento y del sensor, esto se va a poder utilizar desde la oficina donde se encuentre el cliente, el cliente debe estar conectado en la Red LAN de la Universidad. Para poder tener acceso remotamente desde el cliente hacia el servidor, en el servidor debe estar corriendo el servicio de acceso remoto y el cliente va a poder acceder a este por medio de un ID y una contraseña. Tanto en el cliente como en el servidor deben tener instalados y corriendo el software que da servicio al acceso remoto.

3.2.6.2 Servidor de base de datos

El servidor de la base de datos va a estar instalado en el equipo de cómputo servidor, este se va a encargar de guardar todos los datos de temperatura, fecha y hora adecuadamente para poder hacer búsquedas más profundas de los datos, para que esto pueda funcionar el servidor de la base de datos se debe configurar adecuadamente, sí el usuario lo desea se le puede agregar una contraseña de acceso, una vez dentro del servidor se va a tener que crear una base de datos y dentro de esa base de datos contendrán tablas, las que el usuario desee y necesite, dentro de cada tabla se configuraran campos los cuales van a ir almacenados los datos, los campos a configurar podrían ser fecha, hora y temperatura, el servidor de base de datos, va a tener que estar enlazado a la aplicación en Java y también a una plataforma web.

3.2.6.3 Servidor web

Este servidor se va a utilizar para poder almacenar páginas web y así poder mostrarlas en un navegador, las páginas web que se van a crear y van a estar almacenadas dentro del servidor web van a tener que estar conectadas al servidor de base de datos. Todos los equipos de cómputo que se encuentren dentro de la misma Red LAN que el servidor web, van a poder observar las páginas web que contiene dicho servidor. Para poder visualizar la página web, estas van a tener que estar dentro de la carpeta htdocs.

3.2.6.4 Servidor FTP

Este servidor va a funcionar para poder hacer transferencias de archivos remotamente, en el servidor, se va a tener que estar corriendo el servidor FTP y en el cliente se va a tener que estar corriendo un software el cual va a estar conectado remotamente con el servidor, las configuraciones que se le hacen al servidor FTP son muy sencillas, agregar un usuario, una contraseña y una carpeta a compartir, dentro de la carpeta a compartir ahí estarán todos los archivos que se van a transferir, ahora para poder hacer conexión entre el cliente y el servidor, el cliente debe poder acceder correctamente el usuario y la contraseña configurada en el servidor FTP, una vez validado, ahora el cliente va a poder transferir todos los archivos que desee a su equipo de cómputo.

3.2.7 Realizar pruebas de verificación

Una vez que se realizaron todos los pasos anteriores como la verificación de la comunicación LAN, creación del circuito del sensor, creación de los software a utilizar y la instalación y configuración de los servidores, ahora lo que prosigue es realizar las pruebas y verificación de todo el sistema para la medición de temperatura remota de un interferómetro de Sagnac dentro de una Red LAN.

Una vez verificado que todo funcione correctamente y como se tiene previsto, ahora se realizarán las mediciones verdaderas del sistema óptico para poder observar y estudiar cómo se comporta la señal de salida de un interferómetro de Sagnac conforme se le agrega o se le quita temperatura.

3.3 Resumen

En este capítulo se describieron todos los elementos que conforman el arreglo experimental utilizado para resolver la problemática planteada en el capítulo uno.

Dicho arreglo experimental fue implementado principalmente por un cliente y servidores de acceso remoto, de base de datos, web y FTP; los cuales, de manera conjunta permiten recibir la variable física procesada por el Arduino, almacenarla en una base de datos y permitir consultas remotas en un tiempo posterior o tiempo real por parte del cliente. Por otro lado, el cliente fue dotado de una interfaz web, mediante la cual el usuario es capaz de enviar peticiones al servidor correspondiente (web), y de esta manera le pueda proporcionar una conexión remota para realizar la consulta de la información almacenada en un servidor de base de datos. Finalmente la transferencia de archivos entre el cliente y el servidor se realiza mediante una conexión FTP.

CAPÍTULO 4. RESULTADOS Y ANÁLISIS

4.1 Introducción

En este capítulo se presentarán los principales resultados de este trabajo de tesis ya implementado y con sus resultados finales. Se podrá observar como quedaron los servidores junto con sus configuraciones y de igual manera como quedaron las aplicaciones ya terminadas, para poder resolver el problema planteado en el capítulo uno.

4.2 Resultados

Se recuerda que el objetivo del planteamiento de esta tesis es medir la temperatura a la cual se encuentra sometido un interferómetro de Sagnac para que sea enviada de manera remota dentro de una Red LAN por medio de un dispositivo Arduino Mega 2560.

A continuación se presentarán los resultados obtenidos y que se dividieron en arreglos ópticos, circuito de la conexión del sensor DS18B20 y el código para su programación, configuraciones de los servidores y la creación de las aplicaciones.

4.2.1 Resultado arreglo óptico

El arreglo óptico es la parte esencial del proyecto, ya que de ahí sale la problemática a resolver de esta tesis, el siguiente arreglo óptico que se va a presentar fue sacado de la tesis del Ingeniero en Redes José Carlos Rodríguez Novelo que tiene como nombre, “Caracterización de los Componentes Ópticos en un Sistema que Atenúa Ruido ASE por Medio de un Interferómetro de Sagnac que Funciona como Filtro Pasa Banda” por medio de la Universidad de Quintana Roo y la Universidad Autónoma del Carmen como se puede observar en la figura 18.

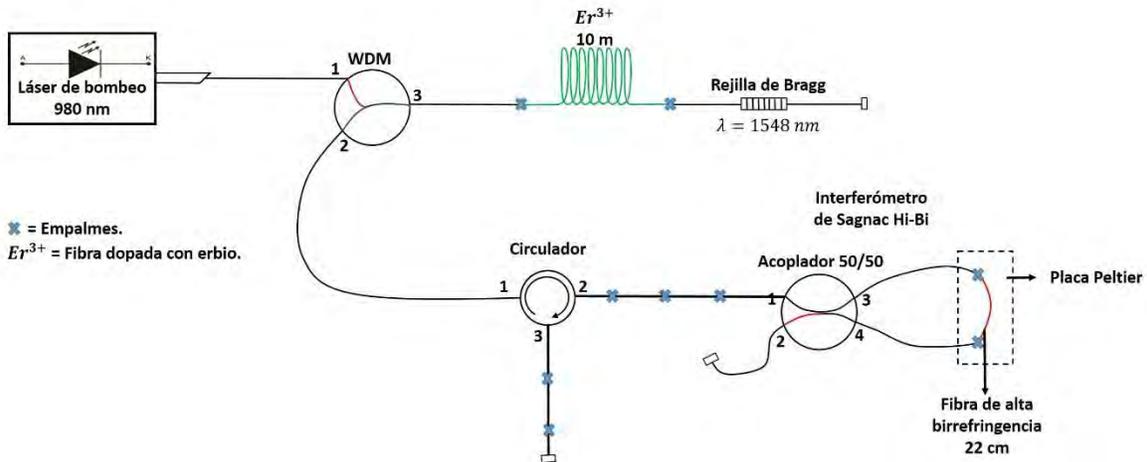


Figura 18. Resultado del arreglo óptico utilizado en la tesis

4.2.2 Resultado circuito del sensor de temperatura DS18B20 con el Arduino MEGA 2560

Para poder medir la temperatura a la cual se encuentra sometido un interferómetro de Sagnac se tuvo que crear un circuito con el cual se estará habilitando el sensor de temperatura DS18B20 y este tuvo que estar conectado al hardware Arduino y así poderse conectar vía serial al servidor. Para poder crear el circuito en donde se encuentra conectado el sensor de temperatura se tuvo que crear una placa el cual contiene dicho circuito. El circuito para el sensor de temperatura se podrá observar en la figura 19.

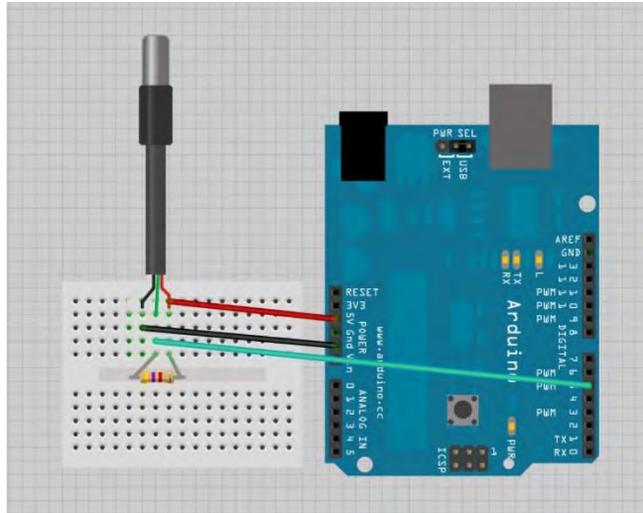


Figura 19. Conexión del Arduino con el sensor DS18B20

Al estar observando el circuito, se realizó una conclusión en donde se podrían anexar más sensores, por medio del software LiveWire y el PCB Wizard, se creó un circuito para crear una placa y así poder conectar seis sensores al mismo tiempo para darse una idea que esta tesis puede incrementarse y no quedarse estancado, esta placa que se creó se llama Shield de temperatura para Arduino, se pueden observar las pistas de la placa en la figura 20.

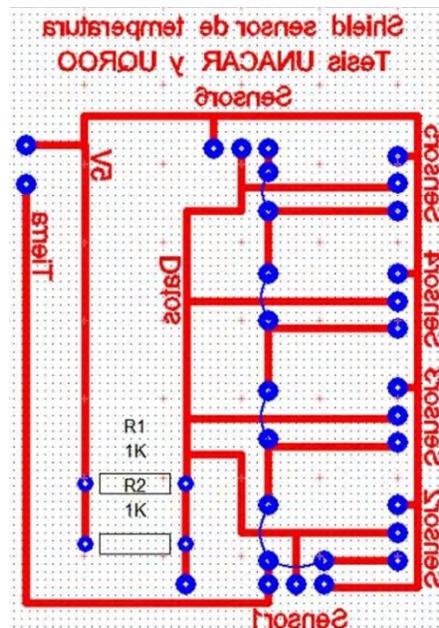


Figura 20. Plantilla del circuito en PCB

Luego de tener el circuito ya terminado, lo que se realizo fue imprimirlo en una hoja con una impresora láser y de esa manera poderlo planchar en la placa fenólica, luego taladrar las pistas donde se conectarán los dispositivos electrónicos y al final soldarlos, se mostrará como quedo el Shield del sensor de temperatura para el hardware Arduino, como se puede ver en la figura 21 y figura 22.

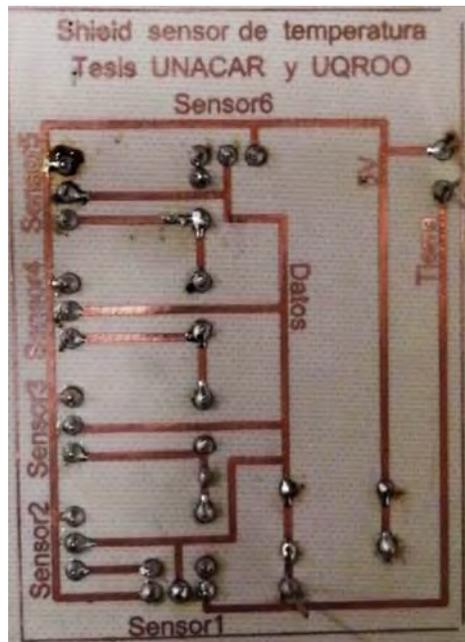


Figura 21. Shield de temperatura para Arduino, creado de forma casera



Figura 22. Shield de temperatura para Arduino, creado de forma casera

4.2.3 Resultado código de temperatura para el Arduino

Una vez creado el Shield de temperatura para el Arduino, se puede conectar el sensor, pero este no podrá enviar los datos de temperatura al Arduino y posteriormente al servidor por medio de un enlace serial, para poder realizar este paso tan importante se tuvo que programar el Arduino MEGA 2560 para poder recibir los datos de temperatura y así poder enviárselo al servidor como se puede observar en la figura 23, a continuación se mostrará el código de programación.

```

#include <OneWire.h> //Se importa la libreria OneWire, con esta librería se puede enviar y recibir
datos por un solo cable
#include <DallasTemperature.h> //Se importa la libreria DallasTemperature para poder hacer la medición del
sensor Dallas DS18B20
#define Pin 40 //Se declara el pin donde se conectará la DATA
OneWire ourWire(Pin); //Se establece el pin declarado como bus para la comunicación OneWire
DallasTemperature sensors(&ourWire); //Se instancia la librería DallasTemperature

boolean key = false; //Se crean variables
int mensaje = 0;
float gc1;
float gc2;

void setup() {
delay(1000); //Se espera un tiempo de 1s
Serial.begin(9600); //Se inicializa la comunicación serial con 9600 baudios
sensors.begin(); //Se inician los sensores
}

void loop() {
sensors.requestTemperatures(); //Prepara el sensor para la lectura

if(Serial.available()>0){ //Se crean sentencias para que se haga la toma de datos,
mensaje = Serial.read(); //cuando mensaje es = a 1 se hace la toma de datos,
if(mensaje=='1'){ //cuando mensaje es = 0 se detiene y no se hace la toma de datos
key = true;
}
else{
key = false;
}
}
if(key == true){ //Si se hace la toma de datos y la llave es verdadera, entonces,
gc1 = sensors.getTempCByIndex(0); //los datos se almacenan en una variable y se imprime en pantalla
Serial.println(gc1, 1); //Se lee e imprime la temperatura en grados Celsius
gc2 = sensors.getTempCByIndex(1);
Serial.println(gc2, 1); //Se lee e imprime la temperatura en grados Celsius

delay(5000); //Se provoca un lapso de 5 segundos antes de la próxima lectura
}
}

```

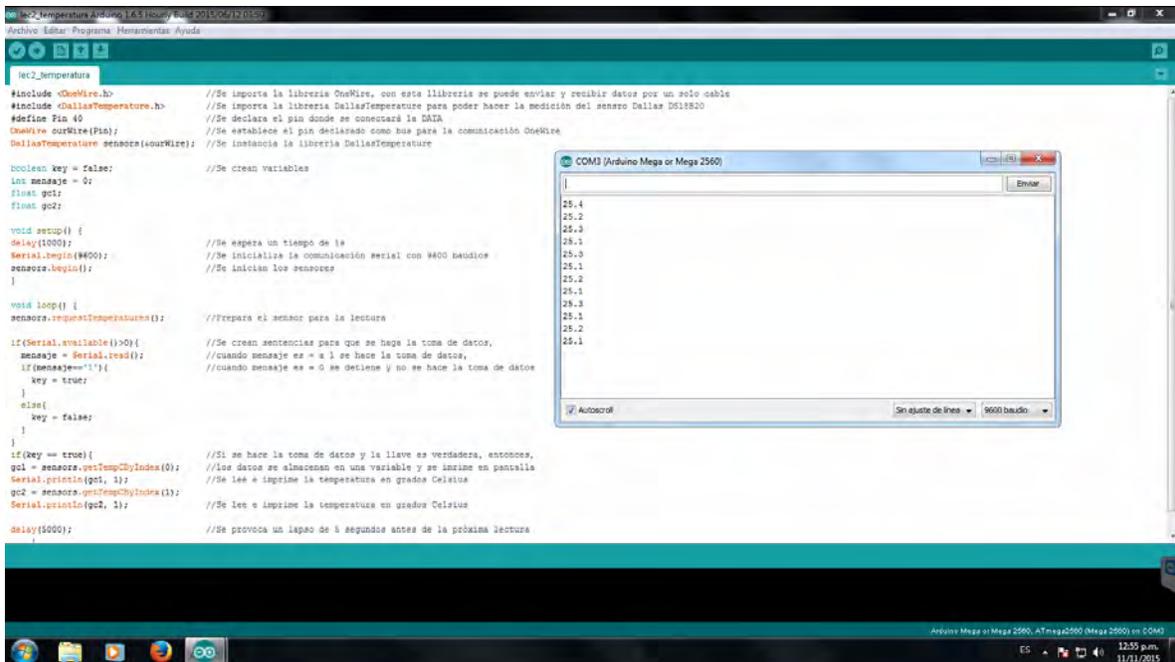


Figura 23. Datos del sensor DS18B20 programados en Arduino

4.2.4 Resultado conexión y configuración de un servidor para un acceso remoto

Conforme se va avanzando, se va resolviendo el objetivo, que es hacer la medición y el monitoreo de la temperatura que se encuentra sometida aun interferómetro de Sagnac para que sea enviada de forma remota dentro de una Red LAN, ahora para realizar el monitoreo de la temperatura, se va a tener que hacer por medio de un acceso remoto.

En el laboratorio donde se ubicó el experimento, se tuvo que agregar una computadora de escritorio, esta computadora de escritorio va a funcionar como servidor y va a estar conectada dentro de la Red LAN de la universidad. En la computadora de escritorio se instaló el software Team Viewer para funcionar como servidor remoto y de igual manera se instaló en la computadora cliente. Para darse una idea de cómo iba a quedar el diseño, se creó un diagrama el cual se encuentra en la figura 24 para poder observar cómo va a estar conectado el cliente y servidor dentro de la Red LAN.

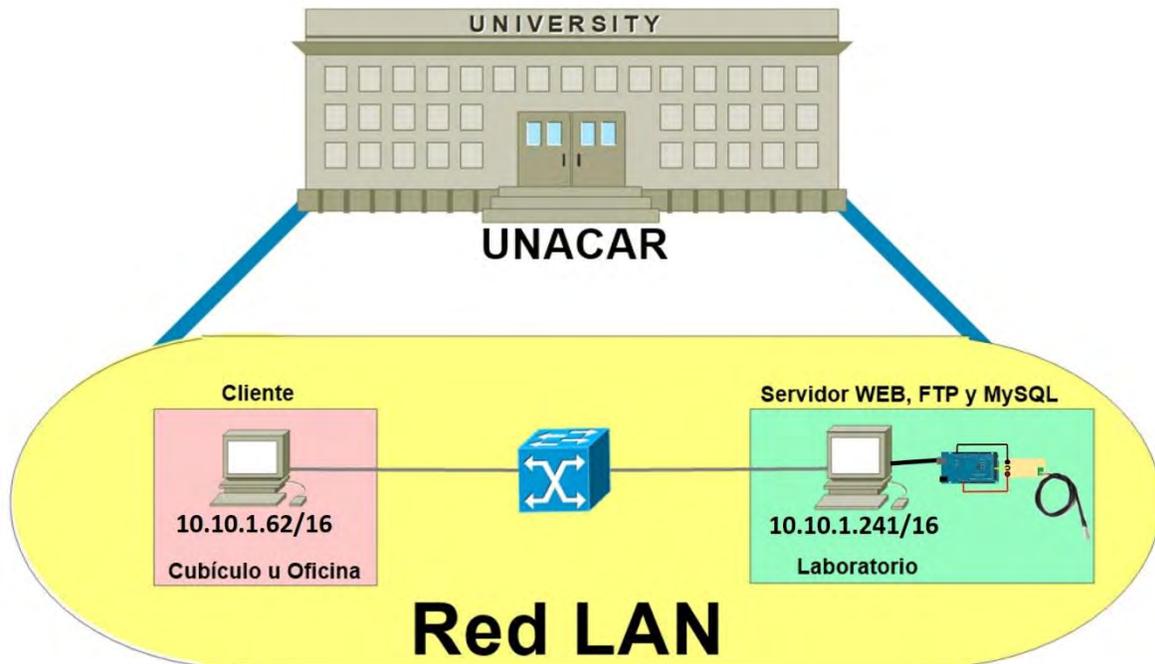


Figura 24. Diagrama de la tesis dentro de una Red LAN

Una vez creado el diagrama, se puso en marcha la instalación del servidor y del cliente, posteriormente se corroboró que el cliente y el servidor se encuentren en comunicación como lo muestra la figura 25 y 26, después el cliente podrá monitorear los datos de temperatura de manera remota como se observa en la figura 27 y 28.

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\MAY ALARCON>ping 10.10.1.241

Haciendo ping a 10.10.1.241 con 32 bytes de datos:
Respuesta desde 10.10.1.241: bytes=32 tiempo=1ms TTL=128
Respuesta desde 10.10.1.241: bytes=32 tiempo<1m TTL=128
Respuesta desde 10.10.1.241: bytes=32 tiempo<1m TTL=128
Respuesta desde 10.10.1.241: bytes=32 tiempo<1m TTL=128

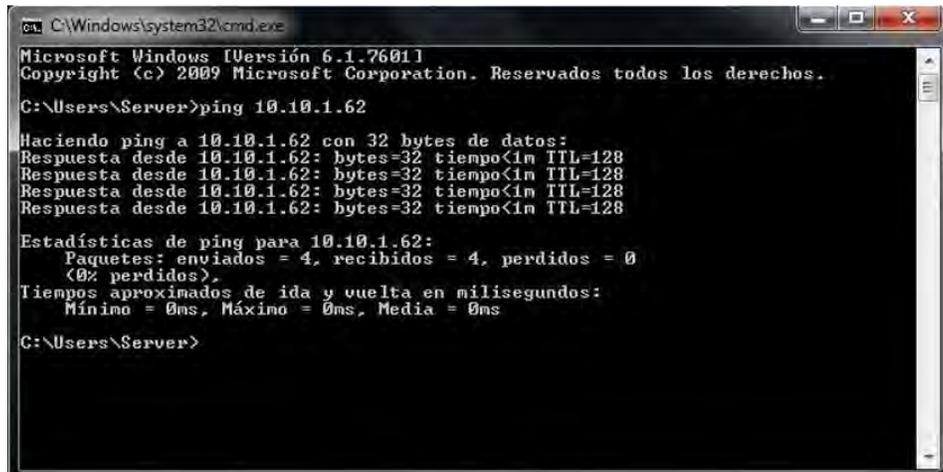
Estadísticas de ping para 10.10.1.241:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms

C:\Users\MAY ALARCON>

```

Figura 25. Conexión cliente – servidor

Verificación de la conexión dentro de la Red LAN del cliente al servidor



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Server>ping 10.10.1.62

Haciendo ping a 10.10.1.62 con 32 bytes de datos:
Respuesta desde 10.10.1.62: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 10.10.1.62:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\Server>
```

Figura 26. Conexión servidor – cliente

Verificación de la conexión dentro de la Red LAN del servidor al cliente

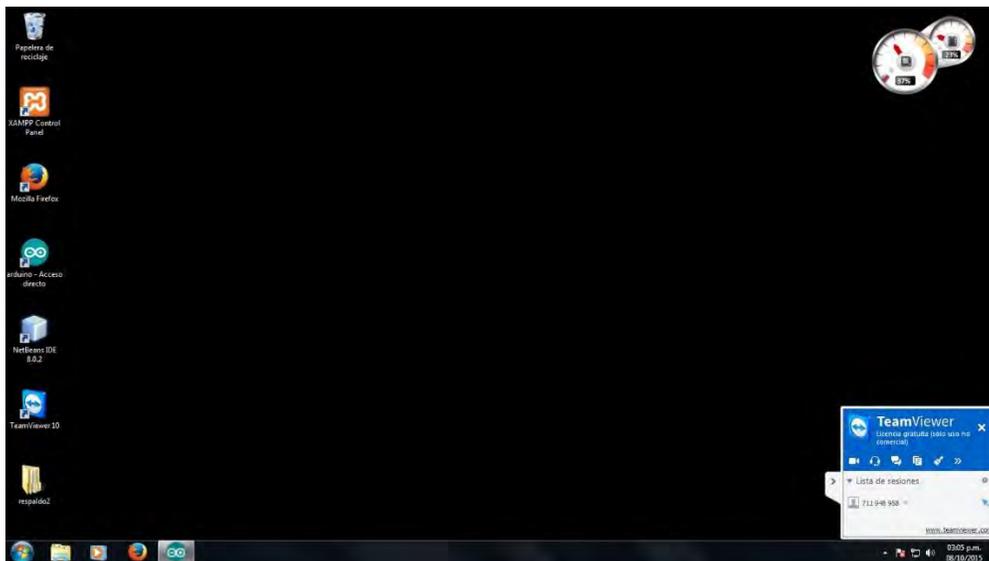


Figura 27. Conexión remota



Figura 28. Monitoreo de la temperatura

4.2.5 Resultado creación de una aplicación con el lenguaje de programación en Java

Ya se puede observar que temperatura arroja el interferómetro de Sagnac, anteriormente la temperatura la proporcionaba de manera muy simple y confusa como se pudo observar en la figura 23, es por eso que se creó una aplicación con el lenguaje de programación en Java para poder observar de una mejor manera la temperatura que contiene el interferómetro de Sagnac junto con la hora y la fecha a la que se hizo dicha captura. También se le agrego un apartado para poder realizar un respaldo de la temperatura por medio de la extensión .xml que es de Excel como se puede observar en la figura 29.

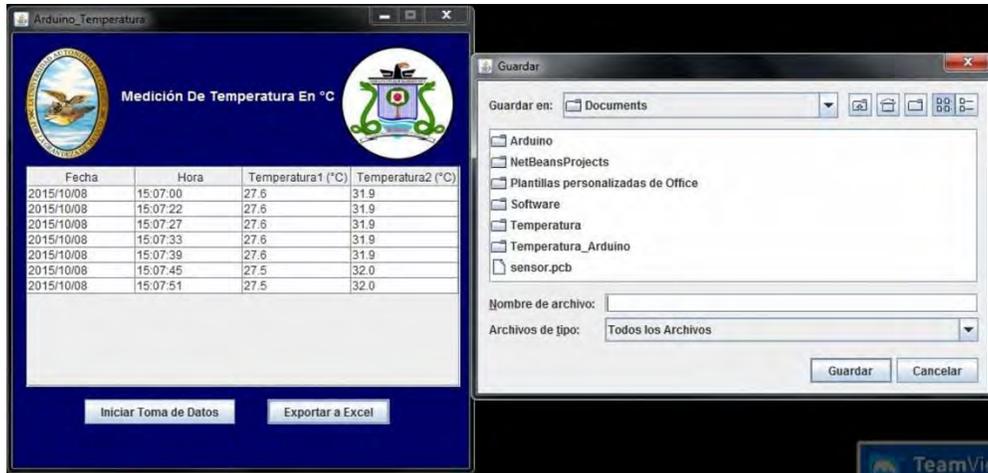


Figura 29. Respaldo de la captura y monitoreo de los datos de temperatura en Excel

4.2.6 Resultado conexión con un servidor FTP para la transferencia de archivos

Ya se puede hacer la captura y el monitoreo de los datos de temperatura de un interferómetro de Sagnac dentro de una Red LAN por medio de un acceso remoto, pero al hacer su dicho respaldo estos datos de temperatura se guardan en el servidor, que es el lugar donde se encuentra la aplicación, es por eso que se tiene que configurar un servidor FTP para poder hacer la transferencia de los archivos de forma remota, de la computadora servidor a la computadora cliente, el resultado de la configuración y del servidor FTP trabajando se puede apreciar en la figura 30 31, 32 y 33.

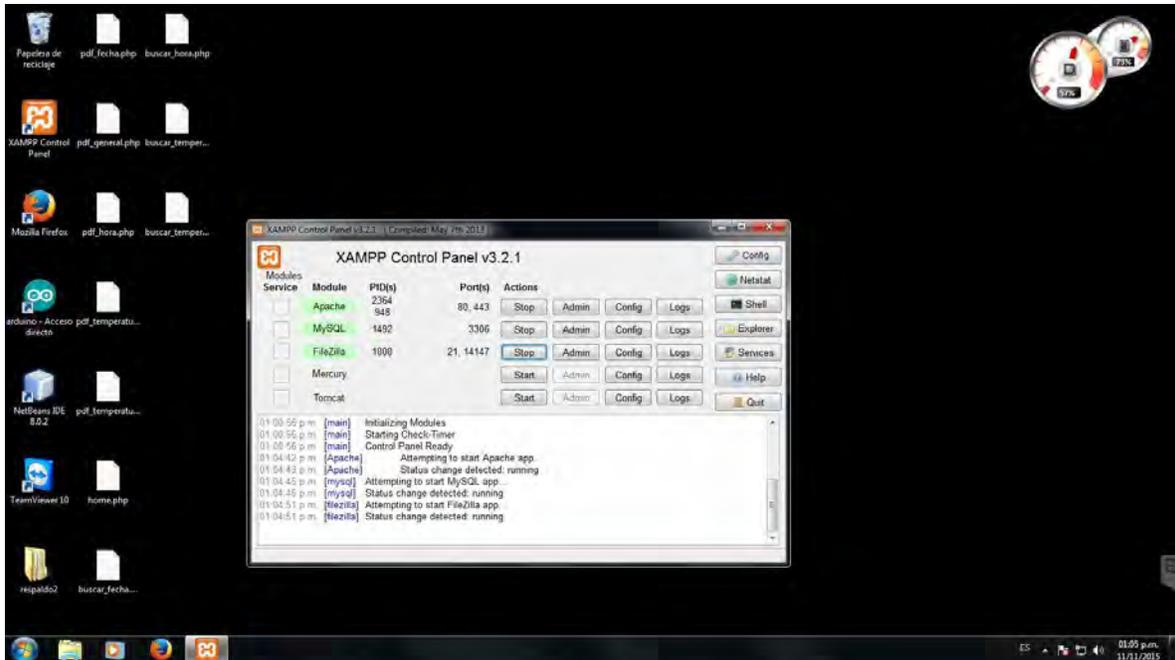


Figura 30. Prendiendo el servidor FTP por medio del software XAMPP

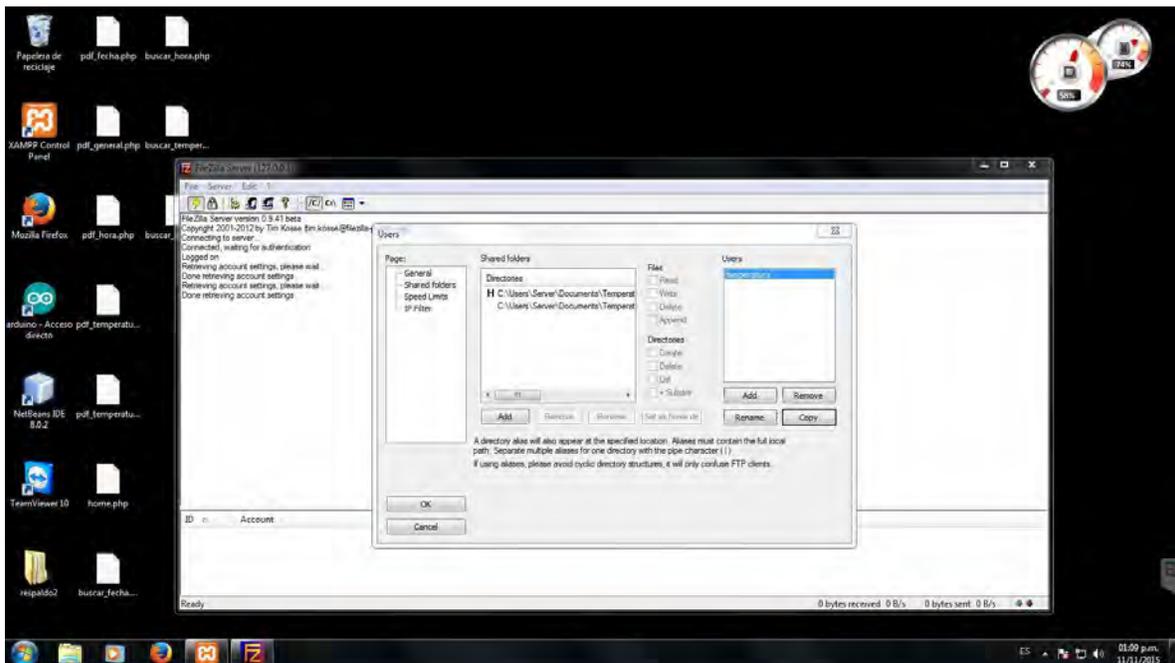


Figura 31. Configurando el servidor FTP con un usuario y una contraseña

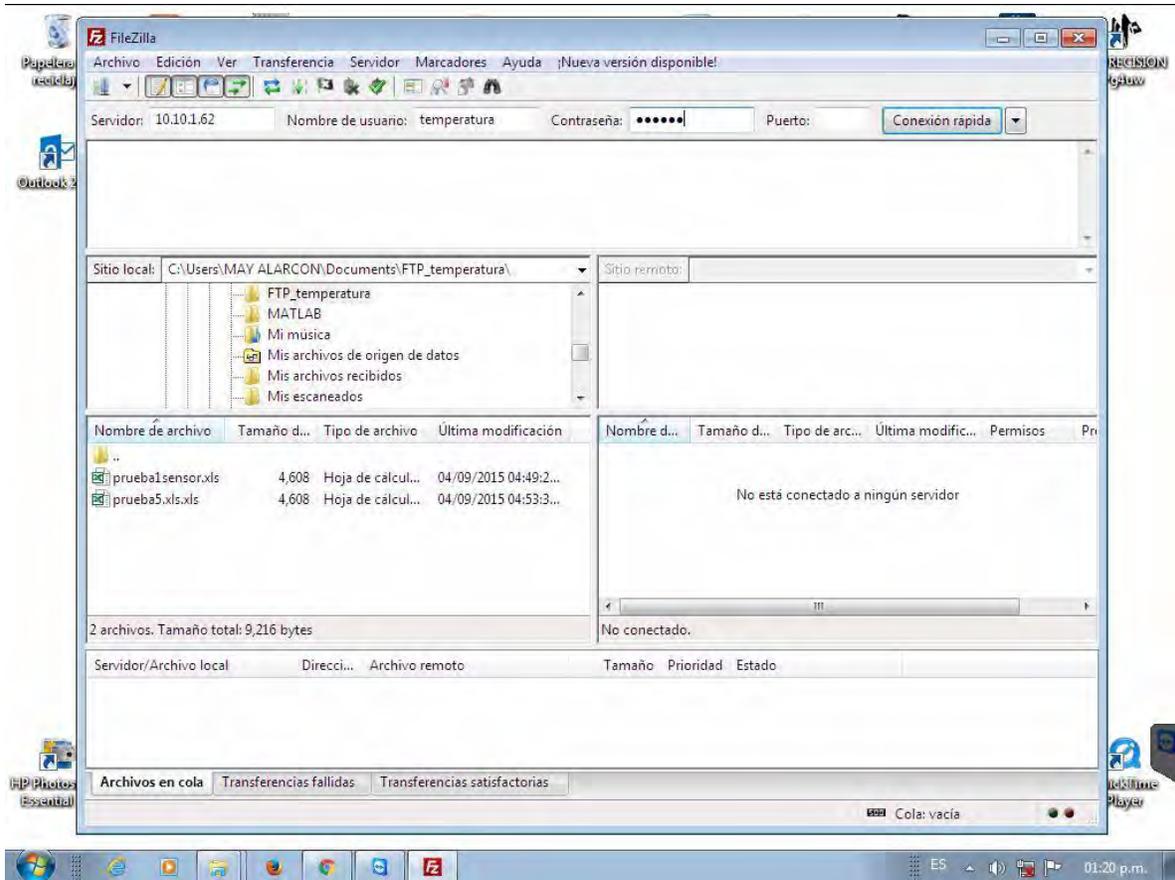


Figura 32. Conectando el servidor FTP desde el cliente hacia el servidor

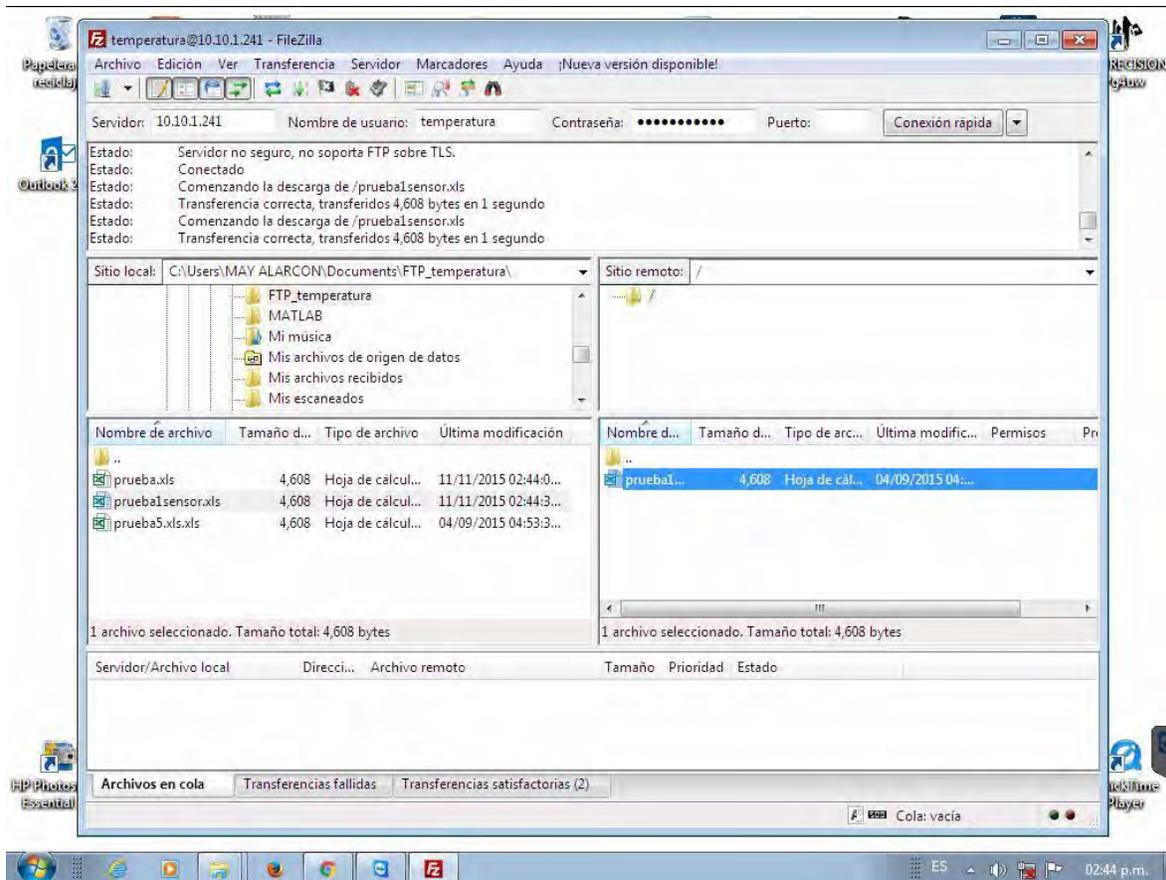


Figura 33. Transferencia de archivos desde el servidor hacia el cliente

4.2.7 Resultado creación de un sistema de búsqueda profunda para los datos de temperatura guardada en una base de datos

Con la aplicación anteriormente ya creada se puede finalizar con esta tesis, conforme se iban haciendo las capturas y el monitoreo de los datos de temperatura, se tenían que ir buscando datos específicos para horas y temperaturas específicas, es por eso que se tuvo que crear un sistema de búsqueda avanzada para los datos de temperatura ya capturados, se tuvo que configurar la plataforma realizada en Java para poder guardar los datos obtenidos en una base de datos automáticamente, y así poder crear un sistema de búsqueda profunda por medio de una plataforma web realizados con HTML5, PHP y SQL.

Por medio de estas etiquetas y lenguajes de programación se creó dicho sistema, con el cual se podrá acceder con un usuario y una contraseña, una vez dentro del

sistema se puede observar que automáticamente se muestran todos los datos guardados junto con su gráfica instantánea, estos datos ya guardados se puede buscar de una manera más compleja por medio de fecha, hora y temperatura, todos estos datos, se pueden guardar en formato PDF y la gráfica mostrada dependiendo de los datos específicos se puede guardar en PDF, JPG y PNG. Este sistema de búsqueda profunda se puede observar en las figuras 34, 35, 36, 37 y 38.



Figura 34. Página principal del sistema de búsquedas



Figura 35. Motor de búsqueda



Figura 36. Cambio de contraseña



Figura 37. Información del contacto

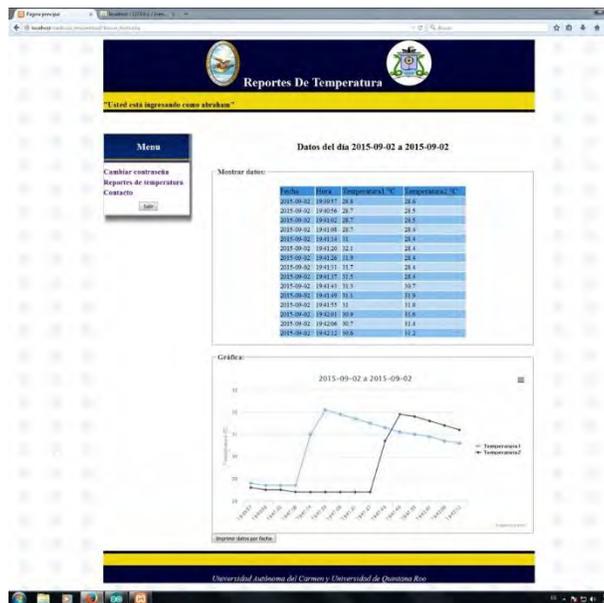


Figura 38. Ejemplo del motor de búsqueda

4.3 Resumen

En este capítulo se presentaron las implementaciones en hardware y software necesarias para obtener el producto final o sistema que permite realizar la detección, medición, almacenamiento y consulta de la temperatura a la que se encuentra expuesta un arreglo óptico. Se mostró el desarrollo del PCB (Printed Circuit Board) o tarjeta de circuito impreso para realizar la conectividad entre los sensores de temperatura y el Arduino; la programación del Arduino para la recepción, procesamiento y la adecuada entrega de los datos de temperatura a un servidor. Además, se presentó la configuración del sistema de comunicación cliente-servidor que permite el acceso remoto a los datos de temperatura almacenados en una base de datos del servidor. Finalmente, se describió el desarrollo de un sistema de búsqueda en la base de datos y desarrollo de la interfaz web de usuario para realizar la visualización y consulta de información de manera fácil, sencilla y amigable.

CAPÍTULO 5. CONCLUSIONES

Los sistemas de comunicaciones ópticos ofrecen numerosas ventajas en la transmisión de información; unas de las más atractivas e importantes son: confiabilidad, aislamiento eléctrico, baja pérdida de transmisión, seguridad, tamaño reducido, bajo peso, y elevado ancho de banda. En estos sistemas, el interferómetro de Sagnac es ampliamente utilizado para eliminar el ruido producido por la Emisión Espontánea Amplificada, entre otras aplicaciones más. Debido a que, la temperatura tiene efectos significativos en la potencia reflejada y potencia transmitida del interferómetro de Sagnac, es ampliamente recomendado, realizar el monitoreo constante de esta variable física.

En este trabajo se presentó la implementación en hardware y software de un sistema que permite realizar la detección, medición, almacenamiento y consulta remota de la variable física temperatura a la que se encuentra expuesto un interferómetro de Sagnac dentro de un arreglo óptico.

El arreglo óptico utilizado para realizar las pruebas del sistema propuesto, corresponde al desarrollado en el trabajo de tesis del Ing. José Carlos Rodríguez Novelo titulado “Caracterización de los Componentes Ópticos de un Sistema que atenúa ruido ASE por medio de un Interferómetro de Sagnac que funciona como Filtro Pasa Banda”.

Para realizar la detección y medición de la variable temperatura en el interferómetro de Sagnac, fue necesario utilizar el sensor de temperatura DS18B20 y el Arduino MEGA 2560. La interconexión entre el sensor y el Arduino se realizó mediante un PCB (Printed Circuit Board) o tarjeta de circuito impreso, la cual fue desarrollada con capacidad para conectar hasta seis sensores de temperatura de manera concurrente.

La programación de la plataforma Arduino fue diseñada para soportar la detección y medición de temperatura hasta de dos sensores de manera paralela; sin embargo, mediante ligeras modificaciones en el código es posible habilitarlo para dar soporte hasta seis sensores de manera concurrente.

El sistema fue desarrollado mediante la arquitectura cliente-servidor y está constituido por un cliente, servidor de acceso remoto, servidor de base de datos, servidor web y servidor FTP; los cuales, de manera conjunta permiten recibir la variable física de temperatura procesada por el Arduino, almacenarla en una base de datos y permitir consultas remotas en un tiempo posterior o tiempo real por parte del cliente. Por otro lado, en el cliente se desarrolló una interfaz web, dicha interfaz web permite al usuario enviar peticiones al servidor web, y establecer una conexión remota para la consulta de la información almacenada en el servidor de base de datos. Finalmente para realizar la transferencia de archivos entre el cliente y el servidor fue necesario utilizar una conexión FTP.

El sistemas propuesto, no solo se puede utilizar para la detección y medición de parámetros en arreglos ópticos, también puede ser utilizado en distintas aplicaciones, tales como: el monitoreo de la temperatura de incubadoras para bebés, el monitoreo de la temperatura dentro del agua, debajo de la tierra y en cualquier superficie, entre muchas otras más.

Una desventajas que presenta este trabajo de tesis es que para incrementar el número de sensores de temperatura a más de dos y máximo seis en el Shield, se tendrá que modificar algunas partes del código en la plataforma Arduino, en la plataforma de la página web y en la plataforma Java. Otra desventaja del proyecto es que si se requiere incrementar el uso de más de seis sensores de temperatura en forma concurrente, será necesario desarrollar otro Shield de temperatura mediante un nuevo PCB o implementar el circuito de manera temporal en un protoboard.

BIBLIOGRAFÍA

- [1] Lackerbauer. (1999). Todo sobre Internet. En I. Lackerbauer, Todo sobre Internet (págs. 192,193, 194, 195). España.
- [2] Hobbs. (1999). *Diseñar su propia página web*. Barcelona: MARCOMBO, pg 13.
- [3] Aufrustein. (2003). Ampliar reparar y configurar su PC (págs. 592). España.
- [4] CONAGUA. (01 de 07 de 2015). Obtenido de CONAGUA: <http://smn.cna.gob.mx/>
- [5] Gómez. Servicios en red, Unidad 3 Servicios de acceso remoto (págs. 65). Editex.
- [6] Herrera. (2003). Tecnologías y redes de transmisión de datos (págs. 121). México: Limusa S.A. de C.V.
- [7] Ibarra-Escamilla. (2001). Teoría del interferómetro de Sagnac de fibra de baja birrefringencia y torcida. *REVISTA MEXICANA DE FÍSICA* 47 (3) 271-274, 1-4.
- [8] X. Fang and R.O. Claus, Optics Letters 20 (1995) 2146.
- [9] D.B. Mortimore, Journal of Lightwave Technology 6 (1988) 1217
- [10] E.A. Kuzin, Optik 106 (1996) 172.
- [11] A.N. Starodumov, Appl. Phys. Lett. 70 (1997) 1.
- [12] Costa, J. (2005). Diccionario de Química Física. En J. Costa, Diccionario de Química Física. España: I Edicions UB.
- [13] Francisco Antonio Ros Gracia. (2010). Guía Práctica de Sensores. En Francisco Antonio Ros Gracia, Guía Práctica de Sensores (pág. 3). España: Creaciones Copyright.
- [14] McRoberts. Beginning Arduino. En M. McRoberts, Beginning Arduino (págs. 2-3).
- [15] Arduino Genuino. (2015). Obtenido de Arduino Genuino: <https://www.arduino.cc/en/Main/Products>.
- [16] LAROUSSE Diccionario Práctico de la Lengua Española. En LAROUSSE Diccionario Práctico de la Lengua Española (págs. 537-538). México: Larousse.

- [17] LAROUSSE Diccionario Práctico de la Lengua Española. En LAROUSSE Diccionario Práctico de la Lengua Española (pág. 717). México: Larousse.
- [18] Caballero. (1998). REDES DE BANDA ANCHA. En Caballero, REDES DE BANDA ANCHA (pág. 111). España: Vanguard.
- [19] Herrera. (2003). Tecnologías de Redes y de Transmisión de Datos. En Herrera, Tecnologías de Redes y de Transmisión de Datos (págs. 121-124). México D.F. LIMUSA.
- [20] Jordi Giera, J. M. (2008). Estructura de Redes de Computadores. En J. M. Jordi Giera, Estructura de Redes de Computadores (págs. 50-51). Barcelona: OUC.

ANEXOS

En este apartado de anexos se incluirán los códigos completos para poder programar el Arduino, el software en Java y la página web. También se incluirán los datasheets antes mencionados en esta tesis.

Código en Arduino para un sensor

```
#include <OneWire.h>           //Se importa la libreria OneWire, con esta libreria se puede enviar y recibir
datos por un solo cable
#include <DallasTemperature.h> //Se importa la libreria DallasTemperature para poder hacer la medición
del sensor Dallas DS18B20
#define Pin 40                 //Se declara el pin donde se conectará la DATA
OneWire ourWire(Pin);         //Se establece el pin declarado como bus para la comunicación OneWire
DallasTemperature sensors(&ourWire); //Se instancia la librería DallasTemperature

boolean key = false;         //Se crean variables
int mensaje = 0;
float gc;

void setup() {
  delay(1000);                //Se espera un tiempo de 1s
  Serial.begin(9600);         //Se inicializa la comunicación serial con 9600 baudios
  sensors.begin();           //Se inician los sensores
}

void loop() {
  sensors.requestTemperatures(); //Prepara el sensor para la lectura

  if(Serial.available()>0){   //Se crean sentencias para que se haga la toma de datos,
    mensaje = Serial.read();   //cuando mensaje es = a 1 se hace la toma de datos,
    if(mensaje=='1'){         //cuando mensaje es = 0 se detiene y no se hace la toma de datos
      key = true;
    }
    else{
      key = false;
    }
  }
  if(key == true){           //Si se hace la toma de datos y la llave es verdadera, entonces,
    gc = sensors.getTempCByIndex(0); //los datos se almacenan en una variable y se imprime en pantalla
    Serial.println(gc, 1);     //Se lee e imprime la temperatura en grados Celsius

    delay(5000);              //Se provoca un lapso de 5 segundos antes de la próxima lectura
  }
}
```

Código en Arduino para dos sensores

```
#include <OneWire.h>           //Se importa la libreria OneWire, con esta libreria se puede enviar y recibir
datos por un solo cable
#include <DallasTemperature.h> //Se importa la libreria DallasTemperature para poder hacer la medición
del sensor Dallas DS18B20
#define Pin 40                 //Se declara el pin donde se conectará la DATA
OneWire ourWire(Pin);         //Se establece el pin declarado como bus para la comunicación OneWire
DallasTemperature sensors(&ourWire); //Se instancia la librería DallasTemperature

boolean key = false;         //Se crean variables
int mensaje = 0;
float gc1;
float gc2;

void setup() {
  delay(1000);                //Se espera un tiempo de un segundo
  Serial.begin(9600);         //Se inicializa la comunicación serial con 9600 baudios
  sensors.begin();           //Se inician los sensores
}

void loop() {
  sensors.requestTemperatures(); //Prepara el sensor para la lectura

  if(Serial.available()>0){    //Se crean sentencias para que se haga la toma de datos,
    mensaje = Serial.read();   //cuando mensaje es = a uno se hace la toma de datos,
    if(mensaje=='1'){          //cuando mensaje es = 0 se detiene y no se hace la toma de datos
      key = true;
    }
    else{
      key = false;
    }
  }
  if(key == true){             //Si se hace la toma de datos y la llave es verdadera, entonces,
    gc1 = sensors.getTempCByIndex(0); //los datos se almacenan en una variable y se imprime en pantalla
    Serial.println(gc1, 1);        //Se lee e imprime la temperatura en grados Celsius
    gc2 = sensors.getTempCByIndex(1);
    Serial.println(gc2, 1);        //Se lee e imprime la temperatura en grados Celsius

    delay(5000);                 //Se provoca un lapso de 5 segundos antes de la próxima lectura
  }
}
```

Código en la plataforma Java para un sensor

Apartado de Java Class

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package gradosarduino;

/**
 *
 * @author Abraham Sierra Calderón
 */
public class GradosArduino {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        //Se instanciar un objeto window para cuando apenas se corra
        //el programa abrirá la ventana
        window ventana = new window();

        //Se le pone una medida fija a la ventana
        ventana.setSize(480,480);
        ventana.setResizable(false);

        //Se hace visible la ventana
        ventana.setVisible(true);
    }
}
```

Apartado del JFrame

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package gradosarduino;

//Se importan las clases a utilizar
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import panamahitek.PanamaHitek_Arduino;
import java.applet.AudioClip;
import java.awt.Color;
import java.sql.*;

/**
 *
 * @author Abraham Sierra Calderón
 */
public class window extends javax.swing.JFrame {

    //Se instancia la libreria Arduino
    PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();

    //Se establece una conexión a la base de datos, declarando unas variables
    Connection con = null;
    PreparedStatement a;

    //Se agrega la URL de la base de datos
    String url = "jdbc:mysql://localhost:3306/1sensor_arduino";

    //Se declara el driver, despues el usuario y password
    String driver = "com.mysql.jdbc.Driver";
    String user = "root";
    String password = "";

    //Se crean variables
    int Slot = 1;
    double Temperaturac = 0;
    int Lecturas = 0;
    boolean State;
    String fechaamd;
    String Output;
}
```

```

int id;

//Se crea una variable de tipo calendario
Calendar Calendario = Calendar.getInstance();

//Metodo para confirmar el cierre del JFrame
public void cerrar() {
    try {
        this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                confirmarSalida();
            }
        });
        this.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//Confirmar salida
public void confirmarSalida() {

    //Se muestran ventanas de advertencias para antes de cerrar por completo
    //el programa
    int valor = JOptionPane.showConfirmDialog(this, "¿Esta seguro de cerrar la aplicación?", "Advertencia",
    JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
    if (valor == JOptionPane.YES_NO_OPTION) {
        JOptionPane.showMessageDialog(null, "Captura finalizada", "Captura finalizada",
    JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}

//Se crea o se instancia un evento con el SerialPortEventListener, cada vez que
//haya un evento serial, se activa el código que se encuentre adentro,
//sirve para la comunicación serial
SerialPortEventListener evento = new SerialPortEventListener() {

    @Override
    public void serialEvent(SerialPortEvent spe) {

        //Cada vez que arduino envíe un salto de línea, se va a mostrar
        //el mensaje enviado por el puerto serial
        if (Arduino.isMessageAvailable()) {

            //Se crea un bucle para que los datos no dejen de enviarse
            //a menos que se requiera
            if (Slot == 1) {

                //cuando la variable sea mayor a 1 se va a activar el método,
                //la variable estará incrementándose después de la lectura de
                //los datos
                if (Lecturas > 1) {
                    TableUpdate();
                }

                Lecturas++;
                Slot = 2;

                //Los datos arrojados por el arduino se encuentran en modo
                //String así que en este paso se convierte a modo double

```

```

        //y se guarda en la variable Temperaturac
        Temperaturac = Double.parseDouble(Arduino.printMessage());
    } else if (Slot == 2) {

        Lecturas++;
        Slot = 1;
    }
}
};

```

```

//Se declara la variable Modelo con la clase DefaultTableModel,
//Esta es una clase que contiene todos los métodos para poder
//modificar datos y añadir filas o columnas en una tabla
DefaultTableModel Modelo;

```

```

//Se crea un método para poder modificar el contenido de la tabla
public void TableUpdate() {

```

```

    if (id > 0) {
        id++;
    }

```

```

//Se instancia un objeto de tipo calendario
GregorianCalendar fecha = new GregorianCalendar();

```

```

//Se crean variables y se inicializan con el año, mes y día
int annio = fecha.get(Calendar.YEAR);
String mes = fecha.get(Calendar.MONTH) + 1 + "";
String dia = fecha.get(Calendar.DAY_OF_MONTH) + "";

```

```

if (Integer.parseInt(mes) < 10) {
    mes = "0" + mes;
}
if (Integer.parseInt(dia) < 10) {
    dia = "0" + dia;
}

```

```

//Se contruye el formato final de la fecha
fechaamd = annio + "/" + mes + "/" + dia;

```

```

//Se inicializa la variable de tipo string
Output = "";

```

```

//Se crean variables y se inicializan con el tipo de
//hora, minuto y segundo
String hora = Calendario.get(Calendar.HOUR_OF_DAY) + "";
String minuto = Calendario.get(Calendar.MINUTE) + "";
String segundos = Calendario.get(Calendar.SECOND) + "";

```

```

//Las variables se convierten a enteros y se condicionan para su
//formato
if (Integer.parseInt(hora) < 10) {
    hora = "0" + hora;
}
if (Integer.parseInt(minuto) < 10) {
    minuto = "0" + minuto;
}
if (Integer.parseInt(segundos) < 10) {
    segundos = "0" + segundos;
}
}

```

```

//En la variable ya se guarda el formato final
Output = hora + ":" + minuto + ":" + segundos;

//Se actualiza el calendario
Calendario = Calendar.getInstance();

//Se agrega la fecha, hora y temperatura a la tabla agregada
Modelo.addRow(new Object[]{fechaamd, Output, Temperaturac});

//Si la temperatura supera la temperatura limite, mostrará un cuadro
//de advertencia!
if (Temperaturac > 34.0) {
    AudioClip alarma;
    alarma = java.applet.Applet.newAudioClip(getClass().getResource("/gradosarduino/alarma.wav"));
    JOptionPane.showMessageDialog(this, "La temperatura incrementó a su límite", "Advertencia", 0);
    //alarma.play();
}

try {

    //Se manda a llamar la variable del driver para la conexión
    //a la base de datos
    Class.forName(driver);

    //Se hace la conexión a la base de datos y se guarda en una variable
    con = DriverManager.getConnection(url, user, password);

    //Se prepara la inserción en SQL y se guarda en una variable
    a = con.prepareStatement("insert into gradosc values(?, ?, ?, ?)");

    //Los datos a guardar se pasan a su formato especificado en la
    //base de datos y se agrega en su campo adecuado
    a.setInt(1, id);
    a.setString(2, fechaamd);
    a.setString(3, Output);
    a.setDouble(4, Temperaturac);

    //Se ejecuta la inserción a la base de datos
    a.executeUpdate();

} catch (ClassNotFoundException e) {
} catch (SQLException e) {
}
}

//Se inicializan los componentes creados en la ventana
public window() {
    initComponents();
    getContentPane().setBackground(new Color(0, 0, 102));
    cerrar();

    //Se inicializa el objeto Modelo, en la tabla agregada, llama
    //jTable1
    Modelo = (DefaultTableModel) jTable1.getModel();

    //Se inicia la conexión con el arduino, con la librería de transmisión y
    //recepción, junto con su COM a utilizar, la velocidad de transferencia y
    //el objeto de la conexión serial
    try {
        Arduino.arduinoRXTX("COM3", 9600, evento);
    } catch (Exception ex) {
        Logger.getLogger(window.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Arduino_Temperatura");

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

        },
        new String [] {
            "Fecha", "Hora", "Temperatura (°C)"
        }
    ));
    jScrollPane1.setViewportView(jTable1);

    jButton1.setText("Iniciar Toma de Datos");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Exportar a Excel");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/gradosarduino/logo_uqroo.png")));
    // NOI18N

    jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/gradosarduino/logo_unacar.png")));
    // NOI18N

    jLabel1.setFont(new java.awt.Font("Arial", 1, 14)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(255, 255, 255));
    jLabel1.setText("Medición De Temperatura En °C");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel3)
            .addContainerGap(10, Short.MAX_VALUE)
        )
    );
}

```

```

        .addGap(20, 20, 20)
        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel2)
        .addGap(42, 42, 42))
    .addGroup(layout.createSequentialGroup())
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
            .addGap(46, 46, 46)
            .addComponent(jButton1)
            .addGap(96, 96, 96)
            .addComponent(jButton2)))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 120,
Short.MAX_VALUE)
                        .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGap(7, 7, 7))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 62,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(36, 36, 36)))
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 230,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addContainerGap(17, Short.MAX_VALUE))
);

pack();
setLocationRelativeTo(null);
} // </editor-fold>

//Se crea un método para poder exportar los datos en Excel
public void ArchivoExcel(String input) throws FileNotFoundException {

    //Se crea un libro, una hoja, una fila y una celda
    HSSFWorkbook libro = new HSSFWorkbook();
    HSSFSheet hoja = libro.createSheet();

    //Las filas y las celdas se crean desde el arreglo (0,0)
    HSSFRow fila = hoja.createRow(0);
    HSSFCell celda = fila.createCell(0);

```

```

//En la celda inicializada se pone el siguiente título
celda.setCellValue("Datos obtenidos del sensor");

//Se crean las celdas junto con sus encabezados
fila = hoja.createRow(1);
celda = fila.createCell(0);
celda.setCellValue("Fecha");
celda = fila.createCell(1);
celda.setCellValue("Hora");
celda = fila.createCell(2);
celda.setCellValue("Temperatura (°C) ");

//Se recorren todos los contenidos de la tabla con el método
//getValue, se pasan a string y se van guardando en las celdas
for (int i = 0; i <= Modelo.getRowCount() - 1; i++) {
    fila = hoja.createRow(i + 2);
    for (int j = 0; j <= 2; j++) {
        celda = fila.createCell(j);
        celda.setCellValue(jTable1.getValueAt(i, j).toString());
    }
}
try {

    //Se guarda el fichero en una ruta
    FileOutputStream Fichero = new FileOutputStream(input);

    //Se genera el fichero
    libro.write(Fichero);

    //Se cierra el fichero
    Fichero.close();
} catch (Exception e) {
    e.printStackTrace();
}
}

//Al hacer click en el botón de importar a Excel, se va a activar
//el código que se encuentre adentro
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    //Aparece una ventana para poder elegir una ruta
    javax.swing.JFileChooser Ventana = new javax.swing.JFileChooser();

    //Se crea una variable string, es donde va a estar la ruta
    String ruta = "";
    try {

        //Aparece la ventana para elegir una ruta y el nombre de donde
        //se encuentra la ruta va a tener la terminación de la
        //extensión .xls que es de Excel
        if (Ventana.showSaveDialog(null) == Ventana.APPROVE_OPTION) {
            ruta = Ventana.getSelectedFile().getAbsolutePath() + ".xls";
            FicheroExcel(ruta);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

//Al hacer click en el botón de iniciar toma de datos, se va a activar
//el código que se encuentre adentro
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

//Estas condiciones, controlan el envío de datos del sensor a la
//aplicación en Java, si el boton esta en un estado verdadero, no
//se hace el envío de datos, de lo contrario se hace el envío de datos
if (State == true) {
    jButton1.setText("Iniciar Toma de Datos");
    State = false;
    try {
        Arduino.sendData("0");
    } catch (Exception ex) {
        Logger.getLogger(window.class.getName()).log(Level.SEVERE, null, ex);
    }
} else {
    State = true;
    jButton1.setText("Terminar Toma de Datos");
    try {
        Arduino.sendData("1");
    } catch (Exception ex) {
        Logger.getLogger(window.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())
        {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new window().setVisible(true);
        }
    });
}

```

```

}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```

Código en la plataforma Java para dos sensores

Apartado Java Class

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package gradosarduino;

/**
 *
 * @author Abraham Sierra Calderón
 */
public class GradosArduino {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        //Se instanciar un objeto window para cuando apenas se corra
        //el programa abrirá la ventana
        window ventana = new window();

        //Se le pone una medida fija a la ventana
        ventana.setSize(480,480);
        ventana.setResizable(false);

        //Se hace visible la ventana
        ventana.setVisible(true);
    }
}

```

Apartado del JFrame

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package gradosarduino;

//Se importan las clases a utilizar
import gnu.io.SerialPortEvent;
import gnu.io.SerialPortEventListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import panamahitek.PanamaHitek_Arduino;
import java.applet.AudioClip;
import java.awt.Color;
import java.sql.*;

/**
 *
 * @author Abraham Sierra Calderón
 */
public class window extends javax.swing.JFrame {

    //Se instancia la libreria Arduino
    PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();

    //Se establece una conexión a la base de datos, declarando unas variables
    Connection con = null;
    PreparedStatement a;

    //Se agrega la URL de la base de datos
    String url = "jdbc:mysql://localhost:3306/2sensores_arduino";

    //Se declara el driver, despues el usuario y password
    String driver = "com.mysql.jdbc.Driver";
    String user = "root";
    String password = "";

    //Se crean variables
    int Slot = 1;
    double Temperaturac = 0;
    double Temperatura2c = 0;
    int Lecturas = 0;
    boolean State;
    String fechaamd;
```

```

String Output;
int id;

//Se crea una variable de tipo calendario
Calendar Calendario = Calendar.getInstance();

//Metodo para confirmar el cierre del JFrame
public void cerrar() {
    try {
        this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                confirmarSalida();
            }
        });
        this.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

//Confirmar salida
public void confirmarSalida() {

    //Se muestran ventanas de advertencias para antes de cerrar por completo
    //el programa
    int valor = JOptionPane.showConfirmDialog(this, "¿Esta seguro de cerrar la aplicación?", "Advertencia",
    JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
    if (valor == JOptionPane.YES_NO_OPTION) {
        JOptionPane.showMessageDialog(null, "Captura finalizada", "Captura finalizada",
    JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);
    }
}

//Se crea o se instancia un evento con el SerialPortEventListener, cada vez que
//haya un evento serial, se activa el código que se encuentre adentro,
//sirve para la comunicación serial
SerialPortEventListener evento = new SerialPortEventListener() {

    @Override
    public void serialEvent(SerialPortEvent spe) {

        //Cada vez que arduino envíe un salto de línea, se va a mostrar
        //el mensaje enviado por el puerto serial
        if (Arduino.isMessageAvailable()) {

            //Se crea un bucle para que los datos no dejen de enviarse
            //a menos que se requiera
            if (Slot == 1) {

                //cuando la variable sea mayor a 1 se va a activar el método,
                //la variable estará incrementándose después de la lectura de
                //los datos
                if (Lecturas > 1) {
                    TableUpdate();
                }

                Lecturas++;
                Slot = 2;

                //Los datos arrojados por el arduino se encuentran en modo

```

```

        //String así que en este paso se convierte a modo souble
        //y se guarda en la variable Temperaturac
        Temperaturac = Double.parseDouble(Arduino.printMessage());
    } else if (Slot == 2) {
        Slot = 1;
        Lecturas++;

        //Los datos arrojados por el arduino se encuentran en modo
        //String así que en este paso se convierte a modo souble
        //y se guarda en la variable Temperatura2c
        Temperatura2c = Double.parseDouble(Arduino.printMessage());
    }
}
};

```

```

//Se declara la variable Modelo con la clase DefaultTableModel,
//Esta es una clase que contiene todos los métodos para poder
//modificar datos y añadir filas o columnas en una tabla
DefaultTableModel Modelo;

```

```

//Se crea un método para poder modificar el contenido de la tabla
public void TableUpdate() {

```

```

    if (id > 0) {
        id++;
    }

```

```

//Se instancia un objeto de tipo calendario
GregorianCalendar fecha = new GregorianCalendar();

```

```

//Se crean variables y se inicializan con el año, mes y día
int annio = fecha.get(Calendar.YEAR);
String mes = fecha.get(Calendar.MONTH) + 1 + "";
String dia = fecha.get(Calendar.DAY_OF_MONTH) + "";

```

```

if (Integer.parseInt(mes) < 10) {
    mes = "0" + mes;
}
if (Integer.parseInt(dia) < 10) {
    dia = "0" + dia;
}

```

```

//Se contruye el formato final de la fecha
fechaamd = annio + "/" + mes + "/" + dia;

```

```

//Se inicializa la variable de tipo string
Output = "";

```

```

//Se crean variables y se inicializan con el tipo de
//hora, minuto y segundo
String hora = Calendario.get(Calendar.HOUR_OF_DAY) + "";
String minuto = Calendario.get(Calendar.MINUTE) + "";
String segundos = Calendario.get(Calendar.SECOND) + "";

```

```

//Las variables se convierten a enteros y se condicionan para su
//formato

```

```

if (Integer.parseInt(hora) < 10) {
    hora = "0" + hora;
}
if (Integer.parseInt(minuto) < 10) {
    minuto = "0" + minuto;
}

```

```

}
if (Integer.parseInt(segundos) < 10) {
    segundos = "0" + segundos;
}

//En la variable ya se guarda el formato final
Output = hora + ":" + minuto + ":" + segundos;

//Se actualiza el calendario
Calendario = Calendar.getInstance();

//Se agrega la fecha, hora y temperatura a la tabla agregada
Modelo.addRow(new Object[]{fechaamd, Output, Temperaturac, Temperatura2c});

//Si la temperatura supera la temperatura limite, mostrará un cuadro
//de advertencia!
if (Temperaturac > 100.0 && Temperatura2c > 100.0) {
    AudioClip alarma;
    alarma = java.applet.Applet.newAudioClip(getClass().getResource("/gradosarduino/alarma.wav"));
    JOptionPane.showMessageDialog(this, "La temperatura1 y temperatura2 incrementaron a su litmie",
"Advertencia", 0);
    //alarma.play();
} else if (Temperaturac > 100.0) {
    AudioClip alarma;
    alarma = java.applet.Applet.newAudioClip(getClass().getResource("/gradosarduino/alarma.wav"));
    JOptionPane.showMessageDialog(this, "La temperatura1 incremento a su litmie", "Advertencia", 0);
    //alarma.play();
} else if (Temperatura2c > 100.0) {
    AudioClip alarma;
    alarma = java.applet.Applet.newAudioClip(getClass().getResource("/gradosarduino/alarma.wav"));
    JOptionPane.showMessageDialog(this, "La temperatura2 incremento a su litmie", "Advertencia", 0);
    //alarma.play();
}

try {

    //Se manda a llamar la variable del driver para la conexión
    //a la base de datos
    Class.forName(driver);

    //Se hace la conexión a la base da datos y se guarda en una variable
    con = DriverManager.getConnection(url, user, password);

    //Se prepara la insercción en SQL y se guarda en una variable
    a = con.prepareStatement("insert into gradosc values(?, ?, ?, ?, ?)");

    //Los datos a guardar se pasan a su formato especificado en la
    //base de datos y se agrega en su campo adecuado
    a.setInt(1, id);
    a.setString(2, fechaamd);
    a.setString(3, Output);
    a.setDouble(4, Temperaturac);
    a.setDouble(5, Temperatura2c);

    //Se ejecuta la insercción a la base de datos
    a.executeUpdate();

} catch (ClassNotFoundException e) {
} catch (SQLException e) {
}
}

```

```

//Se inicializan los componentes creados en la ventana
public window() {
    initComponents();
    getContentPane().setBackground(new Color(0, 0, 102));
    cerrar();

    //Se inicializa el objeto Modelo, en la tabla agregada, llamda
    //jTable1
    Modelo = (DefaultTableModel) jTable1.getModel();

    //Se inicia la conexión con el arduino, con la libreria de transmisión y
    //recepción, junto con su COM a utilizar, la velocidad de tranferencia y
    // el objeto de la conexión serial
    try {
        Arduino.arduinoRXTX("COM3", 9600, evento);
    } catch (Exception ex) {
        Logger.getLogger(window.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Arduino_Temperatura");

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

        },
        new String [] {
            "Fecha", "Hora", "Temperatura1 (°C)", "Temperatura2 (°C)"
        }
    ));
    jScrollPane1.setViewportView(jTable1);

    jButton1.setText("Iniciar Toma de Datos");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Exportar a Excel");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    });
}

```



```

        .addComponent(jScrollPane1,          javax.swing.GroupLayout.PREFERRED_SIZE,      230,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(13, 13, 13)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton1)
            .addComponent(jButton2))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

//Se crea un método para poder exportar los datos en Excel
public void FicheroExcel(String input) throws FileNotFoundException {

    //Se crea un libro, una hoja, una fila y una celda
    HSSFWorkbook libro = new HSSFWorkbook();
    HSSFSheet hoja = libro.createSheet();

    //Las filas y las celdas se crean desde el arreglo (0,0)
    HSSFRow fila = hoja.createRow(0);
    HSSFCell celda = fila.createCell(0);

    //En la celda inicializada se pone el siguiente título
    celda.setCellValue("Datos obtenidos del sensor");

    //Se crean las celdas junto con sus encabezados
    fila = hoja.createRow(1);
    celda = fila.createCell(0);
    celda.setCellValue("Fecha");
    celda = fila.createCell(1);
    celda.setCellValue("Hora");
    celda = fila.createCell(2);
    celda.setCellValue("Temperatura1 (°C) ");
    celda = fila.createCell(3);
    celda.setCellValue("Temperatura2 (°C) ");

    //Se recorren todos los contenidos de la tabla con el método
    //getValue, se pasan a string y se van guardando en las celdas
    for (int i = 0; i <= Modelo.getRowCount() - 1; i++) {
        fila = hoja.createRow(i + 2);
        for (int j = 0; j <= 3; j++) {
            celda = fila.createCell(j);
            celda.setCellValue(jTable1.getValueAt(i, j).toString());
        }
    }
    try {

        //Se guarda el fichero en una ruta
        FileOutputStream Fichero = new FileOutputStream(input);

        //Se genera el fichero
        libro.write(Fichero);

        //Se cierra el fichero
        Fichero.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

//Al hacer click en el botón de importar a Excel, se va a activar
//el código que se encuentre adentro
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    //Aparece una ventana para poder elegir una ruta
    javax.swing.JFileChooser Ventana = new javax.swing.JFileChooser();

    //Se crea una variable string, es donde va a estar la ruta
    String ruta = "";
    try {

        //Aparece la ventana para elegir una ruta y el nombre de donde
        //se encuentra la ruta va a tener la terminación de la
        //extensión .xls que es de Excel
        if (Ventana.showSaveDialog(null) == Ventana.APPROVE_OPTION) {
            ruta = Ventana.getSelectedFile().getAbsolutePath() + ".xls";
            FicheroExcel(ruta);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

//Al hacer click en el botón de iniciar toma de datos, se va a activar
//el código que se encuentre adentro
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    //Estas condiciones, controlan el envío de datos del sensor a la
    //aplicación en Java, si el boton esta en un estado verdadero, no
    //se hace el envío de datos, de lo contrario se hace el envío de datos
    if (State == true) {
        jButton1.setText("Iniciar Toma de Datos");
        State = false;
        try {
            Arduino.sendData("0");
        } catch (Exception ex) {
            Logger.getLogger(window.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    else {
        State = true;
        jButton1.setText("Terminar Toma de Datos");
        try {
            Arduino.sendData("1");
        } catch (Exception ex) {
            Logger.getLogger(window.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels())

```

```

        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(window.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new window().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

```

Código de la página web para un sensor

El código que se presentara a continuación está hecho con etiquetas HTML5, lenguaje de programación PHP, SQL, JavaScript y CS3.

Conexión a la base de datos de MySQL

```
<?php

//Se crea la conexion a MySQL, se selecciona la base de datos a utilizar y se guarda en una variable en PHP
$con = mysql_connect("localhost", "root", "") or die("Error con la conexion a MySQL");
mysql_select_db("1sensor_arduino") or die("Error con la conexion a la Base de Datos");

//Se realizan busquedas en SQL y se guardan en una variable PHP, seguidamente se ejecutan las busquedas
//con la conexion a su base de datos y se guardan en una variable PHP
$busqueda_fecha = "SELECT DISTINCT Fecha FROM gradosc ORDER BY Fecha ASC";
$seleccionar_busqueda_fecha1 = mysql_query($busqueda_fecha, $con);
$seleccionar_busqueda_fecha2 = mysql_query($busqueda_fecha, $con);

$busqueda_temperatura = "SELECT DISTINCT GradosC FROM gradosc ORDER BY GradosC ASC";
$seleccionar_busqueda_temperatura1 = mysql_query($busqueda_temperatura, $con);
$seleccionar_busqueda_temperatura2 = mysql_query($busqueda_temperatura, $con);

$busqueda_hora = "SELECT DISTINCT Hora FROM gradosc ORDER BY Hora ASC";
$seleccionar_busqueda_hora1 = mysql_query($busqueda_hora, $con);
$seleccionar_busqueda_hora2 = mysql_query($busqueda_hora, $con);

$busqueda_general = "SELECT * FROM gradosc ORDER BY Id ASC";
$seleccion_general = mysql_query($busqueda_general);
$seleccion_general1 = mysql_query($busqueda_general);
$seleccion_general2 = mysql_query($busqueda_general);

?>

Se inicializa una sesión para poder entrar y observar el contenido de la página web.
<?php

        //Se inicializa la sesión
        session_start();

//Se evalua que la sesión continúe verificando una de las variables creadas en control.php, cuando esta ya no
coincida
//con su valor inicial se redirige al archivo de salir.php
if (!$_SESSION["autenticado"]) {
        header("Location: salir.php");
}
?>
```

Inicio de sesion

```
<?php

//Se inicializa la sesión
session_start();

//Se destruye la sesión y se redirecciona a la página index.php
session_destroy();
header("Location: index.php");
?>
```

Control de verificación

```
<?php

//Incluyo la página con código de la conexión a la base de datos
include("conexion.php");

//Todos los datos escritos por el usuario en index.php como el usuario y password
//se van a guardar en variables PHP por el método post para poder utilizarlas en
//la búsqueda de la base de datos
$user = $_POST['usuario_txt'];
$pass = $_POST['password_txt'];

//En la variable seleccionar_user se va a guardar la consulta SQL donde va a mostrar los datos del usuario y
password,
//siempre y cuando el usuario haya escrito bien su usuario y su password
$seleccionar_user = "SELECT * FROM usuarios WHERE Usuario = ".$user." AND Pass = ".$pass." ";

//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_user = mysql_query($seleccionar_user, $con) or die ("errorrrrrrr");

//Se recorre el arreglo para poder ejecutar la variable ejecutar_user
if ($verificar_user = mysql_fetch_array($ejecutar_user)) {

    //Se inicia la sesión
    session_start();

    //Se declaran las variables de sesión y se guarda el nombre del usuario en una variable de sesión
    $_SESSION["autenticado"] = true;
    $_SESSION["usuario"] = $_POST["usuario_txt"];

    //Si los datos que agrego el usuario son iguales a la de la base de datos entonces se abre la sesion
    // y se direcciona a la página home.php
    header("Location: home.php");
} else{

    //En caso contrario que los datos agregados por el usuario no coincidan con los datos de la base de
    //datos entonces se redireccionara el error=si y mostrar su respectivo mensaje programado en
    index.php
    header("Location: index.php?error=si"); } ?>
```

Validación de contraseña

```
<?php

//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");

//Todos los datos escritos por el usuario en password.php
//se van a guardar en variables PHP por el método post para poder utilizarlas en
//la búsqueda de la base de datos
$pass_act = $_POST["password_actual_txt"];
$pass_new = $_POST["nuevo_password_txt"];
$ver_pass = $_POST["verificar_password_txt"];

//En la variable seleccionar_pass_act se va a guardar la consulta SQL donde va a mostrar los datos del
usuario,
//siempre y cuando el usuario haya escrito bien su password
$seleccionar_pass_act = "SELECT * FROM usuarios WHERE Pass = ".$pass_act." ";

//En la variable cambiar_pass se va a guardar la consulta SQL donde va a modificar el password del usuario,
//siempre y cuando el usuario este logueado correctamente
$cambiar_pass = "UPDATE usuarios SET Pass = ".$ver_pass." WHERE Usuario =
".$_SESSION["usuario"]." ";

//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_pass_act = mysql_query($seleccionar_pass_act, $con);

//Se recorre el arreglo para poder ejecutar la variable ejecutar_pass_act
if ($verificar = mysql_fetch_array($ejecutar_pass_act)) {

    //Se agrega una condición que dice, si es password nuevo es igual a la verificación del nuevo
password entonces
    //modificame el password actual por el nuevo
    if ($pass_new == $ver_pass) {

        //En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en
la variable,
        //seguidamente se redirecciona a la página programada para mostrar su mensaje
programado en la página
        //password.php
        $ejecutar_cambiar_pass = mysql_query($cambiar_pass, $con);

        header("Location: password.php?pass_cambio=si");

    }else{
        header("Location: password.php?new_pass=si");
    }
}else{
    header("Location: password.php?pass_act=si");
}
?>
```

Index

```
<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

    <!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de
    asentaciones de las palabras y
    el título de la página, esta información el usuario no lo podrá observar al momento de navegar
    sobre la página WEB -->
    <head>

        <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en
        el navegador -->
        <meta charset="utf-8" />

        <!-- Título de la página -->
        <title>Inicio de sesion</title>

        <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
        <link rel="stylesheet" type="text/css" href="css/misestilos.css">
    </head>

    <!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el
    navegador -->
    <body>

        <!-- Encabezado o cabecera del cuerpo -->
        <header>

            <!-- El DIV define una división o una sección de un documento en HTML -->
            <div>

                <!-- Se agrega un título y a sus lados unas imagenes que trabajan como
                logotipos de la página WEB -->
                <h1> Reportes De Temperatura </h1>
            </div>
        </header>

        <!-- Se agrega el menú o la barra de navegación -->
        <nav>

        </nav>

        <!-- Se utiliza para generar secciones en la página WEB, aqui va toda la información
        presentada en la página WEB -->
        <section>

            <!-- Se agrega un formulario para poder validar un usuario, al hacer click en el
            botón, este envía los datos
            que se encuentren en el formulario a la página que hace referencia el action y lo
            envía por un método oculto
            nombrado como post, se agrega una clase, la cual nos sirve para hacer referencia
            en el archivo CSS y así
            poder agregarle un diseño -->
            <form class="centrar" name="autenticacion_frm" action="control.php"
            method="post" enctype="application/x-www-form-urlencoded">
```

<!-- Con código PHP y con el error_reporting se esta diciendo que la
 página WEB solo muestre mensajes de errores,
 avisos y noticias no las va a mostrar. Tambien dentro de PHP se agrego
 un if en el cual se envian variables
 por el método GET y dice que si la variable es igual a si o no este va a
 mostrar su mensaje correspondiente -->

```

<?php
error_reporting(E_ALL ^ E_NOTICE);
if ($_GET["error"]=="si") {
    echo "<br/>";
    echo "<span class = 'error' >Verifica tus datos</span>";
    echo "<br/> <br/>";
}
else{
    echo "<br/>";
    echo "<h1>Introduce tus datos</h1>";
}
?>
  
```

<!-- Con el input se agregan cuadros de formularios se puede observar
 que se agregan cuadros de texto plano,
 cuadros de texto pero con formato en password (No se visualiza en texto
 plano en la página WEB) y
 un boton tipo submit el cual al hacer click sobre ese botón se va ejecutar el
 código quee esté escrito

```

en la página que redirecciona el action -->
<label>Usuario: </label>
<input class="foco" type="text" name="usuario_txt" placeholder="Introduce
tu usuario" required/>
<br/><br/>
<label>Contraseña: </label>
<input class="foco" type="password" name="password_txt"
placeholder="Introduce tu contraseña" required/>
<br/><br/>
<input class="sobre" type="submit" name="enviar_btn" name="Entrar" />
<br/><br/>
</form>
</section>
  
```

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página
 WEB -->

```

<div class="barra_superior">
</div>

<!-- En esta sección se agrega el pie de la página WEB -->
<footer>
    <p >Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>
  
```

Aparecerá el apartado central de los datos junto con sus gráficas y motor de búsqueda.
 <?php

```

//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
?>
  
```

```

<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>
  
```

```

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras,
el título de la página y Scripts para la creación de graficos interactivos, esta información el usuario no lo podrá
observar
al momento de navegar sobre la página WEB -->
<head>

    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />

    <!-- Título de la página -->
    <title>Página principal</title>

    <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
    <link rel="stylesheet" type="text/css" href="css/misestilos.css">

    <!-- Se importan las librerías jquery -->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <style type="text/css">
    ${demo.css}
    </style>

    <!-- Se define la función que creará la gráfica -->
    <script type="text/javascript">
    $(function () {
        $('#container').highcharts({

            //Se le da un título a la gráfica
            title: {
                text: 'Datos generales' ,
                x: -20 //center
            },

            //Se le puede agregar un subtítulo
            subtitle: {
                text: "",
                x: -20
            },

            //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
            //conexión a la base da datos
            xAxis: {
                categories: [

                    <?php
                    $cont=0;
                    $total = mysql_num_rows($seleccion_general1);
                    while ($ver_hora = mysql_fetch_array($seleccion_general1)) {

                        echo "".$ver_hora['Hora']."";
                        $cont++;
                        if($total!=$cont) echo ",";

                    }?>
                    <?php } ?>
                ]
            },

```

```

//Se le agrega un título al eje de la Y
yAxis: {
    title: {
        text: 'Temperatura (°C)'
    },
    plotLines: [{
        value: 0,
        width: 1,
        color: '#808080'
    }]
},

//Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
tooltip: {
    valueSuffix: '°C'
},
legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
},

//Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
//se crea por medio de una búsqueda y conexión a la base da datos
series: [{
    name: 'Temperatura',
    data: [
        <?php
        $cont = 0;
        $total = mysql_num_rows($seleccion_general2);
        while ($ver_temperatura = mysql_fetch_array($seleccion_general2)) {
            $temp = $ver_temperatura['GradosC'];
            echo $temp;
            $cont++;
            if($total != $cont){
                echo ",";
            }
        }
        ?>
        <?php } ?>
    ]
}];
});
</script>

<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>

    <!-- Encabezado o cabecera del cuerpo -->
    <header>

        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>

```

```

        <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos
de la página WEB -->
        <h1> Reportes De Temperatura </h1>
        </div>
        </header>

        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">

                <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
                <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
</p>
        </nav>

        <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
página WEB -->
        <section class="section_home">

                <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
aparte de
la información central, se crea un segmento lateral que contienen links a otras páginas
dentro de la misma
página WEB -->
                <div class="aside">
                        <h2 class="centrar">Menu</h2>
                        <aside class="espacio">
                                <br/>
                                <a href="password.php"><h3>Cambiar contraseña</h3></a>
                                <a href="home.php"><h3>Reportes de temperatura</h3></a>
                                <a href="contacto.php"><h3>Contacto</h3></a>
                                <form class="centrar">
                                        <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
                                </form>
                        </aside>
                </div>

                <!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -
->
                <div class="informacion">

                        <!-- Se le agrega un título central -->
                        <h2 class="centrar">Reportes De Temperatura</h2>

                        <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el
contenido de un formulario, este
formulario se encarga de mostrar los datos de las fechas que se encuentran en la
base de datos y se permiten
elegir los datos para poder realizar una búsqueda profunda dependiendo del dato
elegido, el código de la búsqueda
profunda se puede observar en la página buscar_fecha.php -->
                        <fieldset>
                                <legend><h3>Búsqueda por fecha</h3></legend>
                                <form name="buscar_fecha_frm" action="buscar_fecha.php"
method="post" enctype="application/x-www-form-urlencoded">
                                        <select name="fecha_inicio_slc">
                                                <?php

```

```

                                while ($ejecutar_fecha_inicio =
mysql_fetch_array($seleccionar_busqueda_fecha1)) {
                                    echo "<option>";
                                    echo $ejecutar_fecha_inicio['Fecha'];
                                    echo "</option>";
                                }
                                ?>
                                </select>
                                <select name="fecha_final_slc">
                                <?php
mysql_fetch_array($seleccionar_busqueda_fecha2)) {
                                    while ($ejecutar_fecha_final =
                                        echo "<option>";
                                        echo $ejecutar_fecha_final['Fecha'];
                                        echo "</option>";
                                    }
                                    ?>
                                </select>
                                <input class="sobre" type="submit" name="bucar_fecha"
value="Buscar por fecha" />
                                </form>
                                </fieldset>

                                <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el
contenido de un formulario, este
formulario se encarga de mostrar los datos de las horas que se encuentran en la
base de datos y se permiten
elegir los datos para poder realizar una busqueda profunda dependiendo del dato
elegido, el código de la busqueda
profunda se puede observar en la página buscar_hora.php -->
                                <fieldset>
                                <legend><h3>Busqueda por hora</h3></legend>
                                <form name="buscar_hora_frm" action="buscar_hora.php" method="post"
enctype="application/x-www-form-urlencoded">
                                <select name="hora_inicio_slc">
                                <?php
mysql_fetch_array($seleccionar_busqueda_hora1)) {
                                    while ($hora_inicio =
                                        echo "<option>";
                                        echo $hora_inicio['Hora'];
                                        echo "</option>";
                                    }
                                    ?>
                                </select>
                                <select name="hora_final_slc">
                                <?php
mysql_fetch_array($seleccionar_busqueda_hora2)) {
                                    while ($hora_final =
                                        echo "<option>";
                                        echo $hora_final['Hora'];
                                        echo "</option>";
                                    }
                                    ?>
                                </select>
                                <input class="sobre" type="submit" name="bucar_hora"
value="Buscar por hora" />
                                </form>
                                </fieldset>

                                <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el
contenido de un formulario, este

```

formulario se encarga de mostrar los datos de las temperaturas que se encuentran en la base de datos y se permiten elegir los datos para poder realizar una búsqueda profunda dependiendo del dato elegido, el código de la búsqueda profunda se puede observar en la página buscar_temperatura.php -->

```

<fieldset>
    <legend><h3>Búsqueda por temperatura</h3></legend>
    <form name="buscar_temperatura_frm" action="buscar_temperatura.php"
method="post" enctype="application/x-www-form-urlencoded">
        <select name="temperatura_inicio_slc">
            <?php
                while ($ejecutar_temperatura_inicio =
mysql_fetch_array($seleccionar_búsqueda_temperatura1)) {
                    echo "<option>";
                    echo $ejecutar_temperatura_inicio['GradosC'];
                    echo "</option>";
                }
            ?>
        </select>
        <select name="temperatura_final_slc">
            <?php
                while ($ejecutar_temperatura_final =
mysql_fetch_array($seleccionar_búsqueda_temperatura2)) {
                    echo "<option>";
                    echo $ejecutar_temperatura_final['GradosC'];
                    echo "</option>";
                }
            ?>
        </select>
        <input class="sobre" type="submit" name="bucar_temperatura"
value="Buscar por temperatura" />
    </form>
</fieldset>

```

datos generales dentro de tablas -->

dentro de tablas -->

```

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
que se encuentran en la base de datos, con sus fechas, horas y temperaturas, todo
dentro de tablas -->
<fieldset>
    <legend><h3>Datos generales: </h3></legend>
    <table class="datos">
        <tr class="datos_cabecera">
            <td>Fecha </td>
            <td>Hora </td>
            <td>Temperatura </td>
        </tr>
        <?php
            $cont=1;

            while ($verificar_general =
mysql_fetch_array($seleccion_general)) {
                if ( ($cont % 2) == 0) {
                    echo "<tr class='datos_cuerpo1'>";
                    echo "<td>";
                    echo $verificar_general['Fecha'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_general['Hora'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_general['GradosC'];
                    echo "</td>";
                }
            }
        }
    </table>
</fieldset>

```

```

        echo "</tr>";
    }else{
        echo "<tr class='datos_cuerpo2'>";
        echo "<td>";
        echo $verificar_general['Fecha'];
        echo "</td>";
        echo "<td>";
        echo $verificar_general['Hora'];
        echo "</td>";
        echo "<td>";
        echo $verificar_general['GradosC'];
        echo "</td>";
        echo "</tr>";
    }
    $cont++;
}
?>
</table>
</fieldset>

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Scscript que se encuentra en la cabecera de la página -->
<fieldset>
    <legend><h3>Gráfica: </h3></legend>
    <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
</fieldset>

<!-- En el siguiente formulario se puede observar un botón, este botón al hacer click
en el, nos imprime
del formato en PDF
se puede visuallizar en la página pdf_general.php -->
<form action="pdf_general.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
    <input class="sobre" type="submit" value="Imprimir datos generales"/>
</form>
<br/><br/>
</div>
</section>

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Cambio de contraseña

```
<?php

//Se incluye el inicio de sesión
include("sesion.php");
?>

<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras y
el título de la página, esta información el usuario no lo podrá observar
al momento de navegar sobre la página WEB -->
<head>

    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />

    <!-- Título de la página -->
    <title>Página principal</title>

    <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
    <link rel="stylesheet" type="text/css" href="css/misestilos.css">
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>

    <!-- Encabezado o cabecera del cuerpo -->
    <header>

        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>

            <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos de
la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>

        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">

            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"

        </p>
    </nav>
```

<!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la página WEB -->

```
<section id="section_pass">
```

aparte de la información central, se crea un segmento lateral que contiene links a otras páginas dentro de la misma página WEB -->

```
<div class="aside">
```

```
<h2 class="centrar">Menú</h2>
```

```
<aside class="espacio">
```

```
<br/>
```

```
<a href="password.php"><h3>Cambiar contraseña</h3></a>
```

```
<a href="home.php"><h3>Reportes de temperatura</h3></a>
```

```
<a href="contacto.php"><h3>Contacto</h3></a>
```

```
<form class="centrar">
```

```
<a href="salir.php"><input class="sobre" type="button"
```

```
name="salir_btn" value="Salir" /></a>
```

```
</form>
```

```
</aside>
```

```
</div>
```

<!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -->

```
<div class="informacion">
```

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el contenido de un formulario, este

formulario se encarga de enviar información insertada en los cuadros de texto que están como password y de ahí

verificar que la contraseña sea la correcta y así poder cambiar a una nueva contraseña, el código para la verificación

y cambio de la nueva contraseña se puede observar en la página cambiar_pass.php -->

```
<fieldset>
```

```
<!-- Se le agrega un título central -->
```

```
<legend><h2>Cambiar contraseña: </h2></legend>
```

```
<form class="centrar" name="password_frm" action="cambiar_pass.php" method="post" enctype="application/x-www-form-urlencoded">
```

<!-- Con código PHP y con el error_reporting se está diciendo que la página WEB solo muestre mensajes de errores,

avisos y noticias no las va a mostrar. También dentro de PHP se agregó un if anidado en el cual se envían variables

por el método GET y dice que si la variable es igual a si este va a mostrar su mensaje correspondiente -->

```
<?php
```

```
error_reporting(E_ALL ^ E_NOTICE);
```

```
if ($_GET["pass_act"]=="si") {
```

```
    echo "<span class='error'> Introduzca nuevamente su contraseña actual </span>";
```

```
    echo "<br/> <br/>";
```

```
}elseif ($_GET["new_pass"]=="si") {
```

```
    echo "<span class='error'> Su nueva contraseña es incorrecta, vuelva a agregarla </span>";
```

```
    echo "<br/> <br/>";
```

```
}elseif ($_GET["pass_cambio"]=="si") {
```

```
    echo "<span id='correcto'> Su contraseña ha sido cambiada satisfactoriamente </span>";
```

```
    echo "<br/> <br/>";
```

```

    }
    ?>

    <!-- Se agregan los campos del formulario, son cuadros de texto
con estilo de password este dice que todo texto escrito
se va a mostrar con puntos y no con caracteres, toda la información
escrita en los cuadros de texto, se van a enviar a su
página ya descrita en el form, al hacer click en el botón de tipo
submit -->
    <label>Contraseña actual: </label>
    <input class="foco" type="password" name="password_actual_txt"
placeholder="Contraseña actual" required/>
    <br/><br/>
    <label>Nueva contraseña: </label>
    <input class="foco" type="password" name="nuevo_password_txt"
placeholder="Nueva contraseña" required/>
    <br/><br/>
    <label>Verificar contraseña: </label>
    <input class="foco" type="password"
name="verificar_password_txt" placeholder="Verificar contraseña" required/>
    <br/><br/>
    <input class="sobre" type="submit" name="cambiar_password_txt"
value="Cambiar contraseña" />
    </form>
    </fieldset>
    <br/><br/>
    </div>
</section>

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Información del contacto

```
<?php

//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
?>

<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras,
el título de la página y Scripts para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al momento de navegar sobre la página WEB -->
<head>

    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />

    <!-- Título de la página -->
    <title>Página principal</title>

    <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
    <link rel="stylesheet" type="text/css" href="css/misestilos.css">
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador --
>
<body>

    <!-- Encabezado o cabecera del cuerpo -->
    <header>

        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>

            <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos de
la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>

        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">

            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"

        </p>
    </nav>
```

<!-- Se utiliza para generar secciones en la página WEB, aqui va toda la información presentada en la página WEB -->

```
<section class="section_home">
```

aparte de la información central, se crea un segmento lateral que contienen links a otras páginas dentro de la misma página WEB -->

```
<div class="aside">
  <h2 class="centrar">Menu</h2>
  <aside class="espacio">
    <br/>
    <a href="password.php"><h3>Cambiar contraseña</h3></a>
    <a href="home.php"><h3>Reportes de temperatura</h3></a>
    <a href="contacto.php"><h3>Contacto</h3></a>
    <form class="centrar">
      <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
    </form>
  </aside>
</div>
```

<!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -->

```
<div class="informacion">
  <!-- Se le agrega un título central -->
  <h2 class="centrar">Información del contacto</h2>
  <p>
    La información de los contactos, servirán exclusivamente para poder
    contactarlos, para dar un punto de opinion, dar aportes para hacer mejor este proyecto
    o para alguna aclaración.
  </p>
```

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los datos de los contactos, los cuales realizaron e hicieron posible este proyecto de la medición de la temperatura remota dentro de una red LAN -->

```
<fieldset>
  <legend><h3>Universidad Autónoma del Carmen: </h3></legend>
  <p>
    Dr. Manuel May Alarcón
    <br/>
    Correo electrónico1: mmay@pampano.unacar.mx
    <br/>
    Correo electrónico2: manuel_may10@hotmail.com
  </p>
</fieldset>
<fieldset>
  <legend><h3>Universidad de Quintana Roo: </h3></legend>
  <p>
    Dr. Homero Toral Cruz
    <br/>
    Correo electrónico1: htoral@uqroo.edu.mx
    <br/>
    Correo electrónico2: homerotorcruz@gmail.com
```

```

        </p>
        <p>
            Br. Abraham Sierra Calderón
            <br/>
            Correo electrónico: abraham_sierra.c@hotmail.com
        </p>
    </fieldset>
    <br/>
</div>
</section>

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Motor de búsqueda por fecha

```

<?php

//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");

//Todos los datos mostrados en el select en home.php, estas son las fechas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por fecha de la base de datos
$fecha_inicio = $_POST["fecha_inicio_slc"];
$fecha_final = $_POST["fecha_final_slc"];

//En la variable buscar_fecha se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre fechas propuestas por el usuario
$buscar_fecha = "SELECT * FROM gradosc WHERE Fecha BETWEEN ".$fecha_inicio." AND
"."$fecha_final."";

//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_fecha = mysql_query($buscar_fecha, $con);
$ejecutar_fecha1 = mysql_query($buscar_fecha, $con);
$ejecutar_fecha2 = mysql_query($buscar_fecha, $con);
?>

<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de asentaciones
de las palabras,
el título de la página y Scrips para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al mmomento de navegar sobre la página WEB -->
<head>

```

```

<!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
<meta charset="utf-8" />

<!-- Título de la página -->
<title>Página principal</title>

<!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
<link rel="stylesheet" type="text/css" href="css/misestilos.css">

<!-- Se importan las librerías jquery -->
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<style type="text/css">
    ${demo.css}
</style>

<!-- Se define la función que creará la gráfica -->
<script type="text/javascript">
$(function () {
    $('#container').highcharts({
        //Se le da un título a la gráfica
        title: {
            text: <?php echo "".$fecha_inicio.""; ?> + ' a ' + <?php echo
"".$fecha_final.""; ?>,
            x: -20 //center
        },
        //Se le puede agregar un subtítulo
        subtitle: {
            text: "",
            x: -20
        },
        //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
        //conexión a la base da datos
        xAxis: {
            categories: [
                <?php
                $cont=0;
                $total = mysql_num_rows($ejecutar_fecha1);
                while ($ver_hora = mysql_fetch_array($ejecutar_fecha1)) {
                    echo "".$ver_hora['Hora']."";
                    $cont++;
                    if($total!=$cont) echo ",";
                }
                ?>
            ]
        },
        yAxis: {
            //Se le agrega un título al eje de la Y
            title: {
                text: 'Temperatura (°C)'
            },
            plotLines: [{
                value: 0,
                width: 1,

```

```

        color: '#808080'
    }
},

//Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
tooltip: {
    valueSuffix: '°C'
},
legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
},

//Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
//se crea por medio de una búsqueda y conexión a la base da datos
series: [{
    name: 'Temperatura',
    data: [
        <?php
        $cont = 0;
        $total = mysql_num_rows($ejecutar_fecha2);
        while ($ver_temperatura = mysql_fetch_array($ejecutar_fecha2)) {
            $temp = $ver_temperatura['GradosC'];
            echo $temp;
            $cont++;
            if($total != $cont){
                echo ",";
            }
        }
        ?>
        <?php } ?>
    ]
}
]);
</script>

<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>

    <!-- Encabezado o cabecera del cuerpo -->
    <header>

        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>

            <!-- Se agrega un título y a sus lados unas imágenes que trabajan como logotipos de
            la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
            </header>

            <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
            código CSS

```

```

y asi poder darle estilo a la barra de navegaci3n-->
<nav class="nav_home">

    <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la p3gina con c3digo CSS
y asi poder darle estilo al nombre del usuario logueado -->
    <p class="sesion"> "Usted est3 ingresando como <?php echo $_SESSION["usuario"]; ?>"
</p>
</nav>

    <!-- Se utiliza para generar secciones en la p3gina WEB, aqui va toda la informaci3n presentada en la
p3gina WEB -->
    <section class="section_home">

aparte de
de la misma
p3gina WEB -->
        <div class="aside">
            <h2 class="centrar">Menu</h2>
            <aside class="espacio">
                <br/>
                <a href="password.php"><h3>Cambiar contrasea</h3></a>
                <a href="home.php"><h3>Reportes de temperatura</h3></a>
                <a href="contacto.php"><h3>Contacto</h3></a>
                <form class="centrar">
                    <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
                </form>
            </aside>
        </div>

        <!-- Dentro de este DIV va contenida toda la informaci3n visible y central de la p3gina WEB --
>
        <div class="informacion">

            <!-- Se le agrega un t3tulo central -->
            <h2 class="centrar">Datos del d3a <?php echo $fecha_inicio; ?> a <?php echo
$fecha_final; ?></h2>

            <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
datos generados entre el
sus fechas, horas y
rangos de fechas que se encuentran en la base de datos descritas por el usuario , con
temperaturas, todo dentro de tablas -->
            <fieldset>
                <legend><h3>Mostrar datos: </h3></legend>
                <table class="datos">
                    <tr class="datos_cabecera">
                        <td>Fecha </td>
                        <td>Hora </td>
                        <td>Temperatura </td>
                    </tr>
                    <?php
                    $c = 1;
                    while ($verificar_fecha = mysql_fetch_array($ejecutar_fecha)) {
                        if ( ($c % 2) == 0) {
                            echo "<tr class='datos_cuerpo1'>";
                            echo "<td>";
                            echo $verificar_fecha['Fecha'];
                            echo "</td>";

```

```

        echo "<td>";
        echo $verificar_fecha['Hora'];
        echo "</td>";
        echo "<td>";
        echo $verificar_fecha['GradosC'];
        echo "</td>";
        echo "</tr>";
    }else{
        echo "<tr class='datos_cuerpo2'>";
        echo "<td>";
        echo $verificar_fecha['Fecha'];
        echo "</td>";
        echo "<td>";
        echo $verificar_fecha['Hora'];
        echo "</td>";
        echo "<td>";
        echo $verificar_fecha['GradosC'];
        echo "</td>";
        echo "</tr>";
    }
    $c++;
}
?>
</table>
</fieldset>

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Scscript que se encuentra en la cabecera de la página -->
<fieldset>
    <legend><h3>Gráfica: </h3></legend>
    <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
</fieldset>

<!-- En este formulario se puede observar solo un botón submit el cual se encargará
de mandar información
a la página pdf_fecha.php por el método post, esta página redirigida se encargará de
crear un PDF, la información
a enviar, es el contenido guardado en las variables en PHP descritas al principio y
de tipo hidden para poder
utilizarlas en el documento a redirigir -->
<form action="pdf_fecha.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
    <input type="hidden" id="t" name="fecha_inicio" value = "<?php echo
$fecha_inicio; ?>" />
    <input type="hidden" id="t" name="fecha_final" value = "<?php echo
$fecha_final; ?>" />
    <input class="sobre" type="submit" value="Imprimir datos por fecha" />
</form>
</div>
</section>

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body> </html>

```

Motor de búsqueda por hora

```
<?php

//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");

//Todos los datos mostrados en el select en home.php, estas son las horas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por hora de la base de datos
$inicio = $_POST["hora_inicio_slc"];
$fin = $_POST["hora_final_slc"];

//En la variable buscar_hora se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las horas propuestas por el usuario
$buscar_hora = "SELECT * FROM gradosc WHERE Hora BETWEEN '". $inicio.'" AND '". $fin.'" ";

//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_hora = mysql_query($buscar_hora, $con);
$ejecutar_hora1 = mysql_query($buscar_hora, $con);
$ejecutar_hora2 = mysql_query($buscar_hora, $con);
?>

<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras,
el título de la página y Scripts para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al momento de navegar sobre la página WEB -->
<head>

    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />

    <!-- Título de la página -->
    <title>Página principal</title>

    <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
    <link rel="stylesheet" type="text/css" href="css/misestilos.css">

    <!-- Se importan las librerias jquery -->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <style type="text/css">
    ${demo.css}
    </style>

    <!-- Se define la función que creará la gráfica -->
    <script type="text/javascript">
    $(function () {
        $('#container').highcharts({

                //Se le da un título a la gráfica
                title: {
```

```

                                text: <?php echo "".$inicio."";?> + ' a ' + <?php echo "".$fin."";?> ,
x: -20 //center
},

//Se le puede agregar un subtítulo
subtitle: {
    text: "",
    x: -20
},

//Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
//conexión a la base da datos
xAxis: {
    categories: [

        <?php
        $cont=0;
        $total = mysql_num_rows($ejecutar_hora1);
        while ($ver_hora = mysql_fetch_array($ejecutar_hora1)) {

            echo "".$ver_hora['Hora']."";
            $cont++;
            if($total!=$cont) echo ",";

            ?>
        <?php } ?>
    ]
},

//Se le agrega un título al eje de la Y
yAxis: {
    title: {
        text: 'Temperatura (°C)'
    },
    plotLines: [{
        value: 0,
        width: 1,
        color: '#808080'
    }]
},

//Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
tooltip: {
    valueSuffix: '°C'
},
legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
},

//Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
//se crea por medio de una búsqueda y conexión a la base da datos
series: [{
    name: 'Temperatura',
    data: [
        <?php
        $cont = 0;
        $total = mysql_num_rows($ejecutar_hora2);
        while ($ver_temperatura = mysql_fetch_array($ejecutar_hora2)) {
            $temp = $ver_temperatura['GradosC'];

```

```

        echo $temp;
        $cont++;
        if($total != $cont){
            echo ",";
        }
        ?>
        <?php } ?>
    ]
}
});
</script>

<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>

    <!-- Encabezado o cabecera del cuerpo -->
    <header>

        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>

            <!-- Se agrega un título y a sus lados unas imágenes que trabajan como logotipos
            de la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>

        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
        código CSS
        y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">

            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
            usada en la página con código CSS
            y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
        </p>
        </nav>

        <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
        página WEB -->
        <section class="section_home">

            <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
            aparte de
            la información central, se crea un segmento lateral que contiene links a otras páginas
            dentro de la misma
            página WEB -->
            <div class="aside">
                <h2 class="centrar">Menu</h2>
                <aside class="espacio">
                    <br/>
                    <a href="password.php"><h3>Cambiar contraseña</h3></a>
                    <a href="home.php"><h3>Reportes de temperatura</h3></a>
            </div>
        </section>
    </body>
</html>

```

```

        <a href="contacto.php"><h3>Contacto</h3></a>
        <form class="centrar">
            <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
        </form>
    </aside>
</div>

<!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -
->
<div class="informacion">

    <!-- Se le agrega un título central -->
    <h2 class="centrar">Datos de la hora <?php echo $inicio; ?> a la hora <?php echo
$fin; ?> </h2>

    <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
datos generados entre el
rango de horas que se encuentran en la base de datos descritas por el usuario ,
con sus fechas, horas y
temperaturas, todo dentro de tablas -->
    <fieldset>
        <legend><h3>Mostrar datos: </h3></legend>
        <table class="datos">
            <tr class="datos_cabecera">
                <td>Fecha </td>
                <td>Hora </td>
                <td>Temperatura </td>
            </tr>
            <?php
            $c = 1;
            while ($verificar_hora = mysql_fetch_array($ejecutar_hora)) {
                if ( ($c % 2) == 0) {
                    echo "<tr class='datos_cuerpo1'>";
                    echo "<td>";
                    echo $verificar_hora['Fecha'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_hora['Hora'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_hora['GradosC'];
                    echo "</td>";
                    echo "</tr>";
                }else{
                    echo "<tr class='datos_cuerpo2'>";
                    echo "<td>";
                    echo $verificar_hora['Fecha'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_hora['Hora'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_hora['GradosC'];
                    echo "</td>";
                    echo "</tr>";
                }
                $c++;
            }
            ?>
        </table>
    </fieldset>

```

```

gráfica resalizada por el
auto"></div>
<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
Scrypt que se encuentra en la cabecera de la página -->
<fieldset>
    <legend><h3>Gráfica: </h3></legend>
    <div id="container" style="min-width: 310px; height: 400px; margin: 0
de mandar información
a la página pdf_hora.php por el método post, esta página redirigida se encargará de
crear un PDF, la información
a enviar, es el contenido guardado en las variables en PHP descritas al principio y
de tipo hidden para poder
utilizarlas en el documento a redirigir -->
<form action="pdf_hora.php" method="post" target="_blank" enctype="application/x-
www-form-urlencoded">
    <input type="hidden" id="t" name="hora_inicio" value = "<?php echo
$inicio; ?>"/>
    <input type="hidden" id="t" name="hora_final" value = "<?php echo $fin;
?>"/>
    <input class="sobre" type="submit" value="Imprimir datos por hora"/>
</form>
</div>
</section>

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Motor de búsqueda por temperatura

```
<?php

//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");

//Todos los datos mostrados en el select en home.php, estas son las temperaturas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por temperatura de la base de datos
$temperatura_inicio = $_POST["temperatura_inicio_slc"];
$temperatura_final = $_POST["temperatura_final_slc"];

//En la variable buscar_temperatura se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las temperaturas propuestas por el usuario
$buscar_temperatura = "SELECT * FROM gradosc WHERE GradosC BETWEEN ".$temperatura_inicio." AND
".$temperatura_final." ORDER BY GradosC ASC";

//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_temperatura = mysql_query($buscar_temperatura, $con);
$ejecutar_temperatura1 = mysql_query($buscar_temperatura, $con);
$ejecutar_temperatura2 = mysql_query($buscar_temperatura, $con);
?>

<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>

<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">

<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de asentaciones
de las palabras,
el título de la página y Scrips para la creación de gráficos interactivos, esta información el usuario no lo podrá
observar
al momento de navegar sobre la página WEB -->
<head>

    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />

    <!-- Título de la página -->
    <title>Página principal</title>

    <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
    <link rel="stylesheet" type="text/css" href="css/misestilos.css">

    <!-- Se importan las librerías jquery -->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <style type="text/css">
    ${demo.css}
    </style>

    <!-- Se define la función que creará la gráfica -->
    <script type="text/javascript">
    $(function () {
        $('#container').highcharts({

                //Se le da un título a la gráfica
```

```

        title: {
            text: <?php echo $temperatura_inicio; ?> + ' °C a ' + <?php echo
$temperatura_final; + ' °C' ?>,
            x: -20 //center
        },

//Se le puede agregar un subtítulo
subtitle: {
    text: "",
    x: -20
},

//Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
//conexión a la base da datos
xAxis: {
    categories: [

        <?php
        $cont=0;
        $total = mysql_num_rows($ejecutar_temperatura1);
        while ($ver_hora = mysql_fetch_array($ejecutar_temperatura1)) {

            echo "".$ver_hora['Hora']."";
            $cont++;
            if($total!=$cont) echo ",";

            ?>
            <?php } ?>
        ]
    },

//Se le agrega un título al eje de la Y
yAxis: {
    title: {
        text: 'Temperatura (°C)'
    },
    plotLines: [{
        value: 0,
        width: 1,
        color: '#808080'
    }]
},

//Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
tooltip: {
    valueSuffix: '°C'
},
legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
},

//Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
//se crea por medio de una búsqueda y conexión a la base da datos
series: [{
    name: 'Temperatura',
    data: [
        <?php
        $cont = 0;
        $total = mysql_num_rows($ejecutar_temperatura2);

```

```

        while ($ver_temperatura = mysql_fetch_array($ejecutar_temperatura2)) {
            $temp = $ver_temperatura['GradosC'];
            echo $temp;
            $cont++;
            if($total != $cont){
                echo ",";
            }
            ?>
            <?php } ?>
        ]
    }}
});
</script>

<!-- Se importan las librerias de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>

    <!-- Encabezado o cabecera del cuerpo -->
    <header>

        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>

            <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos de
la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>

        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">

            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
        </p>
        </nav>

        <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
página WEB -->
        <section class="section_home">

            <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
aparte de
de la misma
página WEB -->
            <div class="aside">
                <h2 class="centrar">Menu</h2>
                <aside class="espacio">

```

```

        <br/>
        <a href="password.php"><h3>Cambiar contraseña</h3></a>
        <a href="home.php"><h3>Reportes de temperatura</h3></a>
        <a href="contacto.php"><h3>Contacto</h3></a>
        <form class="centrar">
            <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
        </form>
    </aside>
</div>

<!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB --
>
<div class="informacion">

    <!-- Se le agrega un título central -->
    <h2 class="centrar">Datos de la temperatura <?php echo $temperatura_inicio; ?> °C
a <?php echo $temperatura_final; ?> °C </h2>

    <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
datos generados entre el
rango de temperaturas que se encuentran en la base de datos descritas por el
usuario , con sus fechas, horas y
temperaturas, todo dentro de tablas -->
    <fieldset>
        <legend><h3>Mostrar datos: </h3></legend>
        <table class="datos">
            <tr class="datos_cabecera">
                <td>Fecha </td>
                <td>Hora </td>
                <td>Temperatura </td>
            </tr>
            <?php
            $c = 1;
            while ($verificar_temperatura =
mysql_fetch_array($ejecutar_temperatura)) {
                if ( ($c % 2) == 0) {
                    echo "<tr class='datos_cuerpo1'>";
                    echo "<td>";
                    echo $verificar_temperatura['Fecha'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_temperatura['Hora'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_temperatura['GradosC'];
                    echo "</td>";
                    echo "</tr>";
                }else{
                    echo "<tr class='datos_cuerpo2'>";
                    echo "<td>";
                    echo $verificar_temperatura['Fecha'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_temperatura['Hora'];
                    echo "</td>";
                    echo "<td>";
                    echo $verificar_temperatura['GradosC'];
                    echo "</td>";
                    echo "</tr>";
                }
                $c++;
            }
        }
    }

```

```

    }
    ?>
  </table>
</fieldset>

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Script que se encuentra en la cabecera de la página -->
<fieldset>
  <legend><h3>Gráfica: </h3></legend>
  <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
</fieldset>

<!-- En este formulario se puede observar solo un botón submit el cual se encargará
de mandar información
a la página pdf_temperatura.php por el método post, esta página redirigida se
encargará de crear un PDF, la información
a enviar, es el contenido guardado en las variables en PHP descritas al principio y
de tipo hidden para poder
utilizarlas en el documento a redirigir -->
<form action="pdf_temperatura.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
  <input type="hidden" id="t" name="temperatura_inicio" value = "<?php echo
$temperatura_inicio; ?>" />
  <input type="hidden" id="t" name="temperatura_final" value = "<?php echo
$temperatura_final; ?>" />
  <input class="sobre" type="submit" value="Imprimir datos por
temperatura" />
</form>
</div>
</section>

<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
  <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Respaldo en formato PDF por medio de fecha

```
<?php

//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");

//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por fecha de la base de datos
$fecha_inicio = $_POST['fecha_inicio'];
$fecha_final = $_POST['fecha_final'];

//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{

    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){

        $this->Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this->SetFont('Arial', 'B', 20);
        $this->Cell(163, 10, "Datos ordenados por fecha", 0, 0, 'C');
        $this->Image("pdf/logo_uqroo.jpg", 150, 10, 25, 25);
        $this->Ln(30);
    }
}

//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura"
);

//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();

//Se agrega una página en blanco y se guarda en una variable
$mipdf->addPage();

//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {

    $mipdf->SetFont('Courier', 'B', 15);
    $mipdf->SetTextColor(255, 255, 255);
    $mipdf->Cell(55, 10, $cabeceraT[$i], 1, 0, 'C', true);
}

//Se crea un salto de linea y se guarda en una variable
$mipdf->Ln(10);

//En la variable mostrarr_fecha se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las fechas propuestas por el usuario
$mostrar_fecha = "SELECT * FROM gradosc WHERE Fecha BETWEEN '". $fecha_inicio.'" AND '". $fecha_final.'"
";

//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_fecha = mysql_query($mostrar_fecha, $con);
```

```

//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_fecha = mysql_fetch_array($ejecutar_fecha)) {

    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
    $fecha = $verificar_fecha['Fecha'];
    $hora = $verificar_fecha['Hora'];
    $temperatura = $verificar_fecha['GradosC'];

    //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
    $mipdf -> SetFillColor(100, 100, 200);
    $mipdf -> Cell(55, 10, $fecha, 1, 0, 'C', true);
    $mipdf -> Cell(55, 10, $hora, 1, 0, 'C', true);
    $mipdf -> Cell(55, 10, $temperatura, 1, 0, 'C', true);
    $mipdf -> Ln(10);
}

//Se imprimen todos los datos
$mipdf -> Output();
?>

PDF datos en general
<?php

//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");

//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{

    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){

        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(163, 10, "Datos generales", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 150, 10, 25, 25);
        $this -> Ln(30);
    }
}

//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura"
);

//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();

//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();

//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {

    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
}

```

```

        $mipdf -> Cell(55, 10, $cabeceraT[$i], 1, 0, 'C', true);
    }

    //Se crea un salto de linea y se guarda en una variable
    $mipdf -> Ln(10);

    //En la variable mostrar_datos_general se va a guardar la consulta SQL donde va a mostrar los datos guardados
    //en la tabla gradosc
    $mostrar_dato_general = "SELECT * FROM gradosc";

    //En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
    $ejecutar_dato_general = mysql_query($mostrar_dato_general, $con);

    //Se recorren todos los datos de la tabla ejecutada con el código SQL
    while ($verificar_dato_general = mysql_fetch_array($ejecutar_dato_general)) {

        //Se mandan a llamar todos los datos de la tabla y se guarda en una variable para posteriormente
        usarla
        $fecha = $verificar_dato_general['Fecha'];
        $hora = $verificar_dato_general['Hora'];
        $temperatura = $verificar_dato_general['GradosC'];

        //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
        $mipdf -> SetFillColor(100, 100, 200);
        $mipdf -> Cell(55, 10, $fecha, 1, 0, 'C', true);
        $mipdf -> Cell(55, 10, $hora, 1, 0, 'C', true);
        $mipdf -> Cell(55, 10, $temperatura, 1, 0, 'C', true);
        $mipdf -> Ln(10);
    }

    //Se imprimen todos los datos
    $mipdf -> Output();
    ?>

```

Respaldo en formato PDF datos por hora

```

<?php

//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");

//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por hora de la base de datos
$hora_inicio = $_POST['hora_inicio'];
$hora_final = $_POST['hora_final'];

//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{

    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){

        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(163, 10, "Datos ordenados por hora", 0, 0, 'C');
    }
}

```

```

        $this -> Image("pdf/logo_uqroo.jpg", 150, 10, 25, 25);
        $this -> Ln (30);
    }
}

//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura"
);

//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();

//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();

//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {

    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(55, 10, $cabeceraT[$i], 1, 0, 'C', true);

}

//Se crea un salto de linea y se guarda en una variable
$mipdf -> Ln(10);

//En la variable mostrar_hora se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las horas propuestas por el usuario
$mostrar_hora = "SELECT * FROM gradosc WHERE Hora BETWEEN ".$hora_inicio." AND ".$hora_final." ";

//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_hora = mysql_query($mostrar_hora, $con);

//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_hora = mysql_fetch_array($ejecutar_hora)) {

    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
    $fecha = $verificar_hora['Fecha'];
    $hora = $verificar_hora['Hora'];
    $temperatura = $verificar_hora['GradosC'];

    //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
    $mipdf -> SetFillColor(100, 100, 200);
    $mipdf -> Cell(55, 10, $fecha, 1, 0, 'C', true);
    $mipdf -> Cell(55, 10, $hora, 1, 0, 'C', true);
    $mipdf -> Cell(55, 10, $temperatura, 1, 0, 'C', true);
    $mipdf -> Ln(10);

}

//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Respaldo en formato PDF por temperatura

```
<?php

//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");

//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por temperatura de la base de datos
$temperatura_inicio = $_POST['temperatura_inicio'];
$temperatura_final = $_POST['temperatura_final'];

//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{

    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){

        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(163, 10, "Datos ordenados por temperatura", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 150, 10, 25, 25);
        $this -> Ln(30);
    }
}

//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura"
);

//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();

//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();

//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {

    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(55, 10, $cabeceraT[$i], 1, 0, 'C', true);
}

//Se crea un salto de linea y se guarda en una variable
$mipdf -> Ln(10);

//En la variable mostrar_temperatura se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las temperaturas propuestas por el usuario
$mostrar_temperatura = "SELECT * FROM gradosc WHERE GradosC BETWEEN ".$temperatura_inicio."
AND ".$temperatura_final." ";

//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_temperatura = mysql_query($mostrar_temperatura, $con);
```

```

//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_temperatura = mysql_fetch_array($ejecutar_temperatura)) {

    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
    $fecha = $verificar_temperatura['Fecha'];
    $hora = $verificar_temperatura['Hora'];
    $temperatura = $verificar_temperatura['GradosC'];

    //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
    $mipdf -> SetFillColor(100, 100, 200);
    $mipdf -> Cell(55, 10, $fecha, 1, 0, 'C', true);
    $mipdf -> Cell(55, 10, $hora, 1, 0, 'C', true);
    $mipdf -> Cell(55, 10, $temperatura, 1, 0, 'C', true);
    $mipdf -> Ln(10);
}

//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Hoja de estilos para la página web

```

/* Se le agrega una imagen de fondo a todo e cuerpo de la página WEB */
body{
    background-image: url(body.gif);
}

/* Se crea una cuadro en el encabezado con un ancho predeterminado
un margen superior predeterminado y centrado */
header{
    display: block;
    margin: auto;
    width: 1100px;
    padding-top: 2px;
}

/* Todos los títulos que se encuentren dentro del encabezado estaran
centrados, con un color predeterminado y un color de fondo predeterminado */
header h1{
    color: #FFFFFF;
    display: block;
    margin: auto;
    background-color: #030337;
    text-align: center;
}

/* El cuadro de navegación contendra un color de fondo predeterminado,
estara centrado y contendra bordes predeterminados */
nav {
    background-color: #EEDB0D;
    padding-top: 30px;
    display: block;
    margin: auto;
    width: 1100px;
    border-top: solid;
    border-color: #000000;
    border-width: 7px;
}

```

```

}

/* En el área de seccion contendra un color de fondo predeterminado,
bordes predeterminados y con margenes centrados */
section{
    background-color: #FFFFFF;
    border-top: solid;
    border-bottom: solid;
    border-width: 7px;
    border-color: #000000;
    display: block;
    margin: auto;
    width: 1100px;
}

/* Se crea una barra superior para puro diseño, con bordes,
color de fondo y centrado */
.barra_superior{
    background-color: #EEDB0D;
    padding-top: 30px;
    display: block;
    margin: auto;
    width: 1100px;
    border-color: #000000;
    border-width: 7px;
}

/* El pie de página contendra un recuadro con ccolor de fondo y
centrado */
footer{
    background-color: #030337;
    color: #FFFFFF;
    display: block;
    margin: auto;
    width: 1100px;
    padding-top: 2px;
}

/* El texto que se encuentre dentro del recuadro del pie de página
estara alineada, contrendra un tamaño espesifico y sera tipo cursiva */
footer p{
    text-align: center;
    font-size: 20px;
    font-style: oblique;
}

/* Mostrará mensajes de error con la palabra desada y con un color de fondo,
en negrita y con un tamaño espesifico */
.error{
    background-color: #FC0000;
    font-weight: bold;
    font-size: 30px;
}

/* Cuando haya una insercción a la base de datos correcta, mostrará un
texto predefinido con un color de fondo, en negrita y con un tamaño
espesifico */
#correcto{
    background-color: #FFFFFF;
    font-weight: bold;
    font-size: 30px;
}

```

```

/* El nombre de inicio de sesión del usuario sera negrita y con un
tamaño específico */
.sesion{
    font-weight: bold;
    font-size: 20px;
}

/* La barra de navegación contendrá un color de fondo, estará centrado
y contendrá bordes */
.nav_home{
    background-color: #EEDB0D;
    padding-top: 0px;
    display: block;
    margin: auto;
    width: 1100px;
    border-top: solid;
    border-color: #000000;
    border-width: 7px;
}

/* La sección tendrá un color de fondo, bordes, estará centrado
y se irá incrementando conforme avance el texto */
.section_home{
    background-color: #FFFFFF;
    border-top: solid;
    border-bottom: solid;
    border-width: 7px;
    border-color: #000000;
    display: block;
    margin-left: auto;
    margin-right: auto;
    margin-top: -20px;
    width: 1100px;
    overflow: hidden;
}

/* La sección para el password contendrá un color de fondo, bordes, estará
centrado e incrementará conforme se le vaya agregando texto */
#section_pass{

    background-color: #FFFFFF;
    border-top: solid;
    border-bottom: solid;
    border-width: 7px;
    border-color: #000000;
    display: block;
    margin-left: auto;
    margin-right: auto;
    margin-top: -20px;
    width: 1100px;
    padding-top: 0px;
    overflow: hidden;
}

/* El menú lateral contendrá una imagen, un tamaño predeterminado,
un margen y se posicionará del lado izquierdo */
.aside{
    background-image: url(aside.jpg);
    background-repeat: no-repeat;
    background-size: 100% 100%;
    color: #FFFFFF;
}

```

```

        float: left;
        width: 230px;
        height: 230px;
        margin-left: 0px;
        margin-top: 5%;
    }

/* Agrega el espacio entre filas de palabras */
.espacio{
    line-height: 0.5em;
}

/* Toda la información central de la página web se posicionara del
lado derecho y contendra un margen superior y derecho */
.informacion{
    float: right;
    margin-top: 5%;
    width: 75%;
    margin-right: 0px;
}

/* Centra el texto */
.centrar{
    text-align: center;
}

/* El pie de página estara centrado y se posicionara hasta de ultimo
en la parte inferior de la página WEB */
.footer_home{
    clear: both;
    display: block;
    margin: auto;
    width: 1100px;
}

/* Todo texto que contenga h1 y h3 contendra el color especificado */
legend h3, h1{
    color: #3E3E3E;
}

/* Elimina todas las decoraciones de un link */
a{
    text-decoration: none;
}

/* Cuano el mouse se posicione sobre un objeto, este cambiara su color de fondo
color de texto y sera negrita */
.sobre: hover{
    background-color: #000000;
    color: #FFFFFF;
    font-weight: bold;
}

/* Cuando un cursos se pisicione en una caja de texto, este cambiara
el color de fondo, el borde del recuadro cambiara y el color del borde
tambien */
.foco: focus{
    background-color: #D4D4D4;
    border-radius: 5px;
    border: solid;
    border-color: #A5A5A5;
}

```

```

/* Todos los datos estaran centrados */
.datos{
    margin: auto;
    width: 60%;
}

/* Los datos de la cabecera contendra un color de fondo y un
tamaño de letra espesifico */
.datos_cabecera{
    background-color: #3B91DD;
    font-size: 20px;
}

/* Los datos del cuerpo1 contiene un color de fondo espesifico */
.datos_cuerpo1{
    background-color: #C0DCF5;
}

/* Los datos del cuerpo1 contiene un color de fondo espesifico */
.datos_cuerpo2{
    background-color: #8AC0F0;
}

```

Código de la página web para dos sensores

En esta ocacion para crear un sistema de búsqueda profunda para los datos de la temperatura, el código es el mismo que se mostro en el apartado de “código de la página web para un sensor”, a diferencia qe hay que cambiar algunas partes del código y en ciertos apartados. A continuación se mostrarán los apartados a los que se le tiene que cambiar el código y se colocara de color rojo el código que se le agrego a cada apartado para que ahora se muestren los datos de la temperatura pero ahora para dos sensores.

Home

```
<?php
//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
?>
<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>
<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">
<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras,
el título de la página y Scripts para la creación de graficos interactivos, esta información el usuario no lo podrá
observar
al momento de navegar sobre la página WEB -->
<head>
<!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el navegador -->
  <meta charset="utf-8" />
  <!-- Título de la página -->
  <title>Página principal</title>
  <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
  <link rel="stylesheet" type="text/css" href="css/misestilos.css">
  <!-- Se importan las librerías jquery -->
  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
  <style type="text/css">
  ${demo.css}
  </style>
  <!-- Se define la función que creará la gráfica -->
  <script type="text/javascript">
  $(function () {
      $('#container').highcharts({
          //Se le da un título a la gráfica
          title: {
              text: 'Datos generales de la Temperatura1 y Temperatura2' ,
              x: -20 //center
          },
          //Se le puede agregar un subtítulo
          subtitle: {
              text: "",
              x: -20
          },
          //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
          //conexión a la base da datos
          xAxis: {
              categories: [
                  <?php
                  $cont=0;
                  $total = mysql_num_rows($seleccion_general1);
                  while ($ver_hora = mysql_fetch_array($seleccion_general1)) {
                      echo "".$ver_hora['Hora']."";
                      $cont++;
                      if($total!=$cont) echo ", ";
                  }
                  <?php } ?>
              ]
          },
          //Se le agrega un título al eje de la Y
          yAxis: {
              title: {
```

```

        text: 'Temperatura (°C)'
    },
    plotLines: [{
        value: 0,
        width: 1,
        color: '#808080'
    }]
},
//Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
tooltip: {
    valueSuffix: '°C'
},
legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
},
//Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
//se crea por medio de una búsqueda y conexión a la base da datos
series: [{
    name: 'Temperatura1',
    data: [
        <?php
        $cont = 0;
        $total = mysql_num_rows($seleccion_general2);
        while ($ver_temperatura = mysql_fetch_array($seleccion_general2)) {
            $temp = $ver_temperatura['Grados1C'];
            echo $temp;
            $cont++;
            if($total != $cont){
                echo ",";
            }
        }?>
        <?php } ?>
    ]),
    {
        name: 'Temperatura2',
        data: [
            <?php
            $cont = 0;
            $total = mysql_num_rows($seleccion_general3);
            while ($ver_temperatura2 = mysql_fetch_array($seleccion_general3)) {
                $temp2 = $ver_temperatura2['Grados2C'];
                echo $temp2;
                $cont++;
                if($total != $cont){
                    echo ",";
                }
            }?>
            <?php } ?>
        ]
    }
    ]
});
</script>
<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>
<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->

```

```

<body>
  <!-- Encabezado o cabecera del cuerpo -->
  <header>
    <!-- El DIV define una división o una sección de un documento en HTML -->
    <div>
      <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos
de la página WEB -->
      <h1> Reportes De Temperatura </h1>
    </div>
  </header>
  <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
  <nav class="nav_home">
    <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
    <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
  </p>
  </nav>
  <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
página WEB -->
  <section class="section_home">
    <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
aparte de
la información central, se crea un segmento lateral que contienen links a otras páginas
dentro de la misma
página WEB -->
    <div class="aside">
      <h2 class="centrar">Menu</h2>
      <aside class="espacio">
        <br/>
        <a href="password.php"><h3>Cambiar contraseña</h3></a>
        <a href="home.php"><h3>Reportes de temperatura</h3></a>
        <a href="contacto.php"><h3>Contacto</h3></a>
        <form class="centrar">
          <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
        </form>
      </aside>
    </div>
    <!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -
->
    <div class="informacion">
      <!-- Se le agrega un título central -->
      <h2 class="centrar">Reportes De Temperatura</h2>
      <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el
contenido de un formulario, este
formulario se encarga de mostrar los datos de las fechas que se encuentran en la
base de datos y se permiten
elegir los datos para poder realizar una búsqueda profunda dependiendo del dato
elegido, el código de la búsqueda
profunda se puede observar en la página buscar_fecha.php -->
      <fieldset>
        <legend><h3>Búsqueda por fecha</h3></legend>
        <form name="buscar_fecha_frm" action="buscar_fecha.php"
method="post" enctype="application/x-www-form-urlencoded">
          <select name="fecha_inicio_slc">
            <?php

```

```

                                while ($ejecutar_fecha_inicio =
mysql_fetch_array($seleccionar_búsqueda_fecha1)) {
                                    echo "<option>";
                                    echo $ejecutar_fecha_inicio['Fecha'];
                                    echo "</option>";
                                }
                                ?>
                                </select>
                                <select name="fecha_final_slc">
                                <?php
mysql_fetch_array($seleccionar_búsqueda_fecha2)) {
                                    while ($ejecutar_fecha_final =
                                    echo "<option>";
                                    echo $ejecutar_fecha_final['Fecha'];
                                    echo "</option>";
                                }
                                ?>
                                </select>
                                <input class="sobre" type="submit" name="bucar_fecha"
value="Buscar por fecha" />
                                </form>
                                </fieldset>
                                <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el
contenido de un formulario, este
                                formulario se encarga de mostrar los datos de las horas que se encuentran en la
base de datos y se permiten
                                elegir los datos para poder realizar una búsqueda profunda dependiendo del dato
elegido, el código de la búsqueda
                                profunda se puede observar en la página buscar_hora.php -->
                                <fieldset>
                                <legend><h3>Búsqueda por hora</h3></legend>
                                <form name="buscar_hora_frm" action="buscar_hora.php" method="post"
enctype="application/x-www-form-urlencoded">
                                <select name="hora_inicio_slc">
                                <?php
mysql_fetch_array($seleccionar_búsqueda_hora1)) {
                                    while ($hora_inicio =
                                    echo "<option>";
                                    echo $hora_inicio['Hora'];
                                    echo "</option>";
                                }
                                ?>
                                </select>
                                <select name="hora_final_slc">
                                <?php
mysql_fetch_array($seleccionar_búsqueda_hora2)) {
                                    while ($hora_final =
                                    echo "<option>";
                                    echo $hora_final['Hora'];
                                    echo "</option>";
                                }
                                ?>
                                </select>
                                <input class="sobre" type="submit" name="bucar_hora"
value="Buscar por hora" />
                                </form>
                                </fieldset>
                                <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, contiene el
contenido de dos formularios, estos
                                formularios se encarga de mostrar los datos de las temperaturas que se encuentran
en la base de datos y se permiten

```

elegir los datos para poder realizar una búsqueda profunda dependiendo del dato elegido, el código de la búsqueda profunda se puede observar en la página buscar_temperatura.php y buscar_temperatura2.php -->

```

<fieldset>
  <legend><h3>Búsqueda por temperatura</h3></legend>
  <form name="buscar_temperatura_frm" action="buscar_temperatura.php"
method="post" enctype="application/x-www-form-urlencoded">
    <h4>Temperatura1 °C</h4>
    <select name="temperatura1_inicio_slc">
      <?php
        while ($ejecutar_temperatura1_inicio =
mysql_fetch_array($seleccionar_búsqueda_temperatura1inicio)) {
          echo "<option>";
          echo
$ejecutar_temperatura1_inicio['Grados1C'];
          echo "</option>";
        }
      ?>
    </select>
    <select name="temperatura1_final_slc">
      <?php
        while ($ejecutar_temperatura1_final =
mysql_fetch_array($seleccionar_búsqueda_temperatura1final)) {
          echo "<option>";
          echo $ejecutar_temperatura1_final['Grados1C'];
          echo "</option>";
        }
      ?>
    </select>
    <input class="sobre" type="submit" name="bucar_temperatura"
value="Buscar por temperatura1" />
  </form>
  <form name="buscar_temperatura2_frm"
action="buscar_temperatura2.php" method="post" enctype="application/x-www-form-urlencoded">
    <h4>Temperatura2 °C</h4>
    <select name="temperatura2_inicio_slc">
      <?php
        while ($ejecutar_temperatura2_inicio =
mysql_fetch_array($seleccionar_búsqueda_temperatura2inicio)) {
          echo "<option>";
          echo
$ejecutar_temperatura2_inicio['Grados2C'];
          echo "</option>";
        }
      ?>
    </select>
    <select name="temperatura2_final_slc">
      <?php
        while ($ejecutar_temperatura2_final =
mysql_fetch_array($seleccionar_búsqueda_temperatura2final)) {
          echo "<option>";
          echo $ejecutar_temperatura2_final['Grados2C'];
          echo "</option>";
        }
      ?>
    </select>
    <input class="sobre" type="submit" name="bucar_temperatura"
value="Buscar por temperatura2" />
  </form>
</fieldset>

```

datos generales
dentro de tablas -->

```

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
que se encuentran en la base de datos, con sus fechas, horas y temperaturas, todo
</fieldset>
<legend><h3>Datos generales: </h3></legend>
<table class="datos">
  <tr class="datos_cabecera">
    <td>Fecha </td>
    <td>Hora </td>
    <td>Temperatura1 °C </td>
    <td>Temperatura2 °C </td>
  </tr>
  <?php
  $cont=1;

  while ($verificar_general =
mysql_fetch_array($seleccion_general)) {

```

```

  if ( ($cont % 2) == 0) {
    echo "<tr class='datos_cuerpo1'>";
    echo "<td>";
    echo $verificar_general['Fecha'];
    echo "</td>";
    echo "<td>";
    echo $verificar_general['Hora'];
    echo "</td>";
    echo "<td>";
    echo $verificar_general['Grados1C'];
    echo "</td>";
    echo "<td>";
    echo $verificar_general['Grados2C'];
    echo "</td>";
    echo "</tr>";

  }else{
    echo "<tr class='datos_cuerpo2'>";
    echo "<td>";
    echo $verificar_general['Fecha'];
    echo "</td>";
    echo "<td>";
    echo $verificar_general['Hora'];
    echo "</td>";
    echo "<td>";
    echo $verificar_general['Grados1C'];
    echo "</td>";
    echo "<td>";
    echo $verificar_general['Grados2C'];
    echo "</td>";
    echo "</tr>";
  }
  $cont++;
}
?>
</table>
</fieldset>

```

gráfica resalizada por el

```

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
Script que se encuentra en la cabecera de la página -->
<fieldset>
  <legend><h3>Gráfica: </h3></legend>
  <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>

```

```

        </fieldset>
        <!-- En el siguiente formulario se puede observar un botón, este botón al hacer click
en el, nos imprime
del formato en PDF
        todos los datos generales pero ahora en un formato PDF, el código para la creación
        se puede visualizar en la página pdf_general.php -->
        <form action="pdf_general.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
            <input class="sobre" type="submit" value="Imprimir datos generales"/>
        </form>
        <br/><br/>
    </div>
</section>

    <!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
    <div class="barra_superior">
    </div>

    <!-- En esta sección se agrega en pie de la página WEB -->
    <footer class="footer_home">
        <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
    </footer>
</body>
</html>

```

Busqueda por fecha

```

<?php
//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
//Todos los datos mostrados en el select en home.php, estas son las fechas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una busqueda por fecha de la base de datos
$fecha_inicio = $_POST["fecha_inicio_slc"];
$fecha_final = $_POST["fecha_final_slc"];
//En la variable buscar_fecha se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre fechas propuestas por el usuario
$buscar_fecha = "SELECT * FROM gradosc WHERE Fecha BETWEEN ".$fecha_inicio." AND
"."$fecha_final." ";
//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_fecha = mysql_query($buscar_fecha, $con);
$ejecutar_fecha1 = mysql_query($buscar_fecha, $con);
$ejecutar_fecha2 = mysql_query($buscar_fecha, $con);
$ejecutar_fecha3 = mysql_query($buscar_fecha, $con);
?>
<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>
<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">
<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de asentaciones
de las palabras,
el título de la página y Scrips para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al momento de navegar sobre la página WEB -->
<head>
    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />
    <!-- Título de la página -->
    <title>Página principal</title>

```

```

<!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
<link rel="stylesheet" type="text/css" href="css/misestilos.css">
<!-- Se importan las librerías jquery -->
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<style type="text/css">
    ${demo.css}
</style>
<!-- Se define la función que creará la gráfica -->
<script type="text/javascript">
    $(function () {
        $('#container').highcharts({
            //Se le da un título a la gráfica
            title: {
                text: <?php echo "".$fecha_inicio.""; ?> + ' a ' + <?php echo
"".$fecha_final.""; ?> ,
                x: -20 //center
            },
            //Se le puede agregar un subtítulo
            subtitle: {
                text: "",
                x: -20
            },
            //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
            //conexión a la base de datos
            xAxis: {
                categories: [
                    <?php
                    $cont=0;
                    $total = mysql_num_rows($ejecutar_fecha1);
                    while ($ver_hora = mysql_fetch_array($ejecutar_fecha1)) {
                        echo "".$ver_hora['Hora']."";
                        $cont++;
                        if($total!=$cont) echo ",";
                    }
                    ?>
                ]
            },
            yAxis: {
                //Se le agrega un título al eje de la Y
                title: {
                    text: 'Temperatura (°C)'
                },
                plotLines: [{
                    value: 0,
                    width: 1,
                    color: '#808080'
                }]
            },
            //Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
            tooltip: {
                valueSuffix: '°C'
            },
            legend: {
                layout: 'vertical',
                align: 'right',
                verticalAlign: 'middle',
                borderWidth: 0
            },
            //Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
            //se crea por medio de una búsqueda y conexión a la base de datos
            series: [{

```

```

name: 'Temperatura1',
data: [
<?php
$cont = 0;
$total = mysql_num_rows($ejecutar_fecha2);
while ($ver_temperatura = mysql_fetch_array($ejecutar_fecha2)) {
    $temp = $ver_temperatura['Grados1C'];
    echo $temp;
    $cont++;
    if($total != $cont){
        echo ",";
    }
    ?>
<?php } ?>
}],
{
    name: 'Temperatura2',
    data: [
    <?php
    $cont = 0;
    $total = mysql_num_rows($ejecutar_fecha3);
    while ($ver_temperatura2 = mysql_fetch_array($ejecutar_fecha3)) {
        $temp2 = $ver_temperatura2['Grados2C'];
        echo $temp2;
        $cont++;
        if($total != $cont){
            echo ",";
        }
        ?>
    <?php } ?>
    ]
    });
</script>
<!-- Se importan las librerias de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>
<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>
    <!-- Encabezado o cabecera del cuerpo -->
    <header>
        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>
            <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos
de la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>
        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">
            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
        </p>
    </nav>

```

<!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la página WEB -->

```
<section class="section_home">
```

<!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido aparte de

la información central, se crea un segmento lateral que contiene links a otras páginas dentro de la misma

página WEB -->

```
<div class="aside">
```

```
<h2 class="centrar">Menu</h2>
```

```
<aside class="espacio">
```

```
<br/>
```

```
<a href="password.php"><h3>Cambiar contraseña</h3></a>
```

```
<a href="home.php"><h3>Reportes de temperatura</h3></a>
```

```
<a href="contacto.php"><h3>Contacto</h3></a>
```

```
<form class="centrar">
```

```
<a href="salir.php"><input class="sobre" type="button"
```

```
name="salir_btn" value="Salir" /></a>
```

```
</form>
```

```
</aside>
```

```
</div>
```

<!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -

->

```
<div class="informacion">
```

<!-- Se le agrega un título central -->

```
<h2 class="centrar">Datos del día <?php echo $fecha_inicio; ?> a <?php echo $fecha_final; ?></h2>
```

<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los datos generados entre el

rango de fechas que se encuentran en la base de datos descritas por el usuario , con sus fechas, horas y

temperaturas, todo dentro de tablas -->

```
<fieldset>
```

```
<legend><h3>Mostrar datos: </h3></legend>
```

```
<table class="datos">
```

```
<tr class="datos_cabecera">
```

```
<td>Fecha </td>
```

```
<td>Hora </td>
```

```
<td>Temperatura1 °C</td>
```

```
<td>Temperatura2 °C</td>
```

```
</tr>
```

```
<?php
```

```
$c = 1;
```

```
while ($verificar_fecha = mysql_fetch_array($ejecutar_fecha)) {
```

```
if ( ($c % 2) == 0) {
```

```
echo "<tr class='datos_cuerpo1'>";
```

```
echo "<td>";
```

```
echo $verificar_fecha['Fecha'];
```

```
echo "</td>";
```

```
echo "<td>";
```

```
echo $verificar_fecha['Hora'];
```

```
echo "</td>";
```

```
echo "<td>";
```

```
echo $verificar_fecha['Grados1C'];
```

```
echo "</td>";
```

```
echo "<td>";
```

```
echo $verificar_fecha['Grados2C'];
```

```
echo "</td>";
```

```
echo "</tr>";
```

```
}else{
```

```
echo "<tr class='datos_cuerpo2'>";
```

```
echo "<td>";
```

```

        echo $verificar_fecha[Fecha];
        echo "</td>";
        echo "<td>";
        echo $verificar_fecha[Hora];
        echo "</td>";
        echo "<td>";
        echo $verificar_fecha[Grados1C];
        echo "</td>";
        echo "<td>";
        echo $verificar_fecha[Grados2C];
        echo "</td>";
        echo "</tr>";
    }
    $c++;
}
?>
</table>
</fieldset>
<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Script que se encuentra en la cabecera de la página -->
<fieldset>
    <legend><h3>Gráfica: </h3></legend>
    <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
</fieldset>
<!-- En este formulario se puede observar solo un botón submit el cual se encargará
de mandar información
a la página pdf_fecha.php por el método post, esta página redirigida se encargará
de crear un PDF, la información
a enviar, es el contenido guardado en las variables en PHP descritas al principio y
de tipo hidden para poder
utilizarlas en el documento a redirigir -->
<form action="pdf_fecha.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
    <input type="hidden" id="t" name="fecha_inicio" value = "<?php echo
$fecha_inicio; ?>" />
    <input type="hidden" id="t" name="fecha_final" value = "<?php echo
$fecha_final; ?>" />
    <input class="sobre" type="submit" value="Imprimir datos por fecha" />
</form>
<br />
</div>
</section>
<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>

<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Busqueda por hora

```
<?php
//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
//Todos los datos mostrados en el select en home.php, estas son las horas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por hora de la base de datos
$inicio = $_POST["hora_inicio_slc"];
$fin = $_POST["hora_final_slc"];
//En la variable buscar_hora se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las horas propuestas por el usuario
$buscar_hora = "SELECT * FROM gradosc WHERE Hora BETWEEN '". $inicio.'" AND '". $fin.'" ";
//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_hora = mysql_query($buscar_hora, $con);
$ejecutar_hora1 = mysql_query($buscar_hora, $con);
$ejecutar_hora2 = mysql_query($buscar_hora, $con);
$ejecutar_hora3 = mysql_query($buscar_hora, $con);
?>
<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>
<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">
<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras,
el título de la página y Scrips para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al momento de navegar sobre la página WEB -->
<head>
<!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
<meta charset="utf-8" />
<!-- Título de la página -->
<title>Página principal</title>
<!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
<link rel="stylesheet" type="text/css" href="css/misestilos.css">
<!-- Se importan las librerias jquery -->
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<style type="text/css">
${demo.css}
</style>
<!-- Se define la función que creará la gráfica -->
<script type="text/javascript">
$(function () {
    $('#container').highcharts({
        //Se le da un título a la gráfica
        title: {
            text: <?php echo "''. $inicio.'";?> + ' a ' + <?php echo "''. $fin.'";?> ,
            x: -20 //center
        },
        //Se le puede agregar un subtítulo
        subtitle: {
            text: "",
            x: -20
        },
        //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
        //conexión a la base da datos
        xAxis: {
            categories: [
```

```

<?php
$cont=0;
$total = mysql_num_rows($ejecutar_hora1);
while ($ver_hora = mysql_fetch_array($ejecutar_hora1)) {
    echo "".$ver_hora['Hora']."";
    $cont++;
    if($total!=$cont) echo ",";
    ?>
<?php } ?>
]
},
//Se le agrega un título al eje de la Y
yAxis: {
    title: {
        text: 'Temperatura (°C)'
    },
    plotLines: [{
        value: 0,
        width: 1,
        color: '#808080'
    }]
},
//Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
tooltip: {
    valueSuffix: '°C'
},
legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
},
//Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
//se crea por medio de una búsqueda y conexión a la base da datos
series: [{
    name: 'Temperatura1',
    data: [
<?php
$cont = 0;
$total = mysql_num_rows($ejecutar_hora2);
while ($ver_temperatura = mysql_fetch_array($ejecutar_hora2)) {
    $temp = $ver_temperatura['Grados1C'];
    echo $temp;
    $cont++;
    if($total != $cont){
        echo ",";
    }
    ?>
<?php } ?>
    ],
    {
        name: 'Temperatura2',
        data: [
<?php
$cont = 0;
$total = mysql_num_rows($ejecutar_hora3);
while ($ver_temperatura2 = mysql_fetch_array($ejecutar_hora3)) {
    $temp2 = $ver_temperatura2['Grados2C'];
    echo $temp2;
    $cont++;
    if($total != $cont){

```

```

        echo ",";
    }
    ?>
<?php } ?>
    ]]
    ]
});
});
</script>
<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>
<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>
    <!-- Encabezado o cabecera del cuerpo -->
    <header>
        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>
            <!-- Se agrega un título y a sus lados unas imágenes que trabajan como logotipos
de la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>
        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">
            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
        </p>
        </nav>
        <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
página WEB -->
        <section class="section_home">
            <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
aparte de
la información central, se crea un segmento lateral que contienen links a otras páginas
dentro de la misma
página WEB -->
            <div class="aside">
                <h2 class="centrar">Menu</h2>
                <aside class="espacio">
                    <br/>
                    <a href="password.php"><h3>Cambiar contraseña</h3></a>
                    <a href="home.php"><h3>Reportes de temperatura</h3></a>
                    <a href="contacto.php"><h3>Contacto</h3></a>
                    <form class="centrar">
                        <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
                    </form>
                </aside>
            </div>
            <!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -
->
            <div class="informacion">
                <!-- Se le agrega un título central -->

```

```

$fin; ?> </h2>
<h2 class="centrar">Datos de la hora <?php echo $inicio; ?> a la hora <?php echo
datos generados entre el
con sus fechas, horas y
temperaturas, todo dentro de tablas -->
<fieldset>
  <legend><h3>Mostrar datos: </h3></legend>
  <table class="datos">
    <tr class="datos_cabecera">
      <td>Fecha </td>
      <td>Hora </td>
      <td>Temperatura1 °C</td>
      <td>Temperatura2 °C</td>
    </tr>
  <?php
  $c = 1;
  while ($verificar_hora = mysql_fetch_array($ejecutar_hora)) {
    if ( ($c % 2) == 0) {
      echo "<tr class='datos_cuerpo1'>";
      echo "<td>";
      echo $verificar_hora['Fecha'];
      echo "</td>";
      echo "<td>";
      echo $verificar_hora['Hora'];
      echo "</td>";
      echo "<td>";
      echo $verificar_hora['Grados1C'];
      echo "</td>";
      echo "<td>";
      echo $verificar_hora['Grados2C'];
      echo "</td>";
      echo "</tr>";
    }else{
      echo "<tr class='datos_cuerpo2'>";
      echo "<td>";
      echo $verificar_hora['Fecha'];
      echo "</td>";
      echo "<td>";
      echo $verificar_hora['Hora'];
      echo "</td>";
      echo "<td>";
      echo $verificar_hora['Grados1C'];
      echo "</td>";
      echo "<td>";
      echo $verificar_hora['Grados2C'];
      echo "</td>";
      echo "</tr>";
    }
    $c++;
  }
  ?>
</table>
</fieldset>
<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Script que se encuentra en la cabecera de la página -->
<fieldset>
  <legend><h3>Gráfica: </h3></legend>
  <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>

```

```

        </fieldset>
de mandar información
        <!-- En este formulario se puede observar solo un botón submit el cual se encargará
crear un PDF, la información
        a la página pdf_hora.php por el método post, esta página redirigida se encargará de
de tipo hidden para poder
        a enviar, es el contenido guardado en las variables en PHP descritas al principio y
utilizarlas en el documento a redirigir -->
        <form action="pdf_hora.php" method="post" target="_blank" enctype="application/x-
www-form-urlencoded">
                <input type="hidden" id="t" name="hora_inicio" value = "<?php echo
$inicio; ?>"/>
                <input type="hidden" id="t" name="hora_final" value = "<?php echo $fin;
?>"/>
                <input class="sobre" type="submit" value="Imprimir datos por hora"/>
        </form>
        <br/>
    </div>
</section>
<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>
<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Busqueda por la temperatura del primer sensor

```

<?php
//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
//Todos los datos mostrados en el select en home.php, estas son las temperaturas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una busqueda por temperatura de la base de datos
$temperatura1_inicio = $_POST["temperatura1_inicio_slc"];
$temperatura1_final = $_POST["temperatura1_final_slc"];
//En la variable buscar_temperatura1 se va a guardar la consulta SQL donde va a mostrar los datos
guardados
//en la tabla gradosc que se encuentren entre las temperaturas del sensor 1 propuestas por el usuario
$buscar_temperatura1 = "SELECT * FROM gradosc WHERE Grados1C BETWEEN ".$temperatura1_inicio."
AND ".$temperatura1_final." ORDER BY Grados1C ASC";
//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_temperatura = mysql_query($buscar_temperatura1, $con);
$ejecutar_temperatura1 = mysql_query($buscar_temperatura1, $con);
$ejecutar_temperatura2 = mysql_query($buscar_temperatura1, $con);
?>
<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>
<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">
<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de asentaciones
de las palabras,
el título de la página y Scrips para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al mmomento de navegar sobre la página WEB -->
<head>

```

```

        <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
        <meta charset="utf-8" />
        <!-- Título de la página -->
        <title>Página principal</title>
        <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
        <link rel="stylesheet" type="text/css" href="css/misestilos.css">
        <!-- Se importan las librerías jquery -->
        <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
        <style type="text/css">
        ${demo.css}
        </style>
        <!-- Se define la función que creará la gráfica -->
        <script type="text/javascript">
        $(function () {
                $('#container').highcharts({
                        //Se le da un título a la gráfica
                        title: {
                                text: <?php echo $temperatura1_inicio; ?> + ' °C a ' + <?php echo
$temperatura1_final; + ' °C' ?> ,
                                x: -20 //center
                        },
                        //Se le puede agregar un subtítulo
                        subtitle: {
                                text: "",
                                x: -20
                        },
                        //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
                        //conexión a la base da datos
                        xAxis: {
                                categories: [

                                        <?php
                                        $cont=0;
                                        $total = mysql_num_rows($ejecutar_temperatura1);
                                        while ($ver_hora = mysql_fetch_array($ejecutar_temperatura1)) {
                                                echo "".$ver_hora['Hora']."";
                                                $cont++;
                                                if($total!=$cont) echo ",";
                                                ?>
                                        <?php } ?>
                                ]
                        },
                        //Se le agrega un título al eje de la Y
                        yAxis: {
                                title: {
                                        text: 'Temperatura (°C)'
                                },
                                plotLines: [{
                                        value: 0,
                                        width: 1,
                                        color: '#808080'
                                }]
                        },
                        //Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
                        tooltip: {
                                valueSuffix: '°C'
                        },
                        legend: {
                                layout: 'vertical',
                                align: 'right',

```

```

        verticalAlign: 'middle',
        borderWidth: 0
    },
    //Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
    //se crea por medio de una búsqueda y conexión a la base da datos
    series: [{
        name: 'Temperatura1',
        data: [
            <?php
            $cont = 0;
            $total = mysql_num_rows($ejecutar_temperatura2);
            while ($ver_temperatura = mysql_fetch_array($ejecutar_temperatura2)) {
                $temp = $ver_temperatura['Grados1C'];
                echo $temp;
                $cont++;
                if($total != $cont){
                    echo ",";
                }
            }?>
            <?php } ?>
        ]
    }
    ]
});
</script>
<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>

<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->
<body>
    <!-- Encabezado o cabecera del cuerpo -->
    <header>
        <!-- El DIV define una división o una sección de un documento en HTML -->
        <div>
            <!-- Se agrega un título y a sus lados unas imágenes que trabajan como logotipos
            de la página WEB -->
            <h1> Reportes De Temperatura </h1>
            </div>
        </header>
        <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
        código CSS
        y así poder darle estilo a la barra de navegación-->
        <nav class="nav_home">
            <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
            usada en la página con código CSS
            y así poder darle estilo al nombre del usuario logueado -->
            <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
        </p>
        </nav>
        <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
        página WEB -->
        <section class="section_home">
            <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
            aparte de
            la información central, se crea un segmento lateral que contiene links a otras páginas
            dentro de la misma
            página WEB -->
            <div class="aside">

```

```

<h2 class="centrar">Menu</h2>
<aside class="espacio">
  <br/>
  <a href="password.php"><h3>Cambiar contraseña</h3></a>
  <a href="home.php"><h3>Reportes de temperatura</h3></a>
  <a href="contacto.php"><h3>Contacto</h3></a>
  <form class="centrar">
    <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
  </form>
</aside>
</div>
<!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -
->
<div class="informacion">
  <!-- Se le agrega un título central -->
  <h2 class="centrar">Datos de la temperatura <?php echo $temperatura1_inicio; ?>
°C a <?php echo $temperatura1_final; ?> °C</h2>
  <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
datos generados entre el
usuario , con sus fechas, horas y
temperaturas, todo dentro de tablas -->
  <fieldset>
    <legend><h3>Mostrar datos: </h3></legend>
    <table class="datos">
      <tr class="datos_cabecera">
        <td>Fecha </td>
        <td>Hora </td>
        <td>Temperatura1 °C</td>
        <td>Temperatura2 °C</td>
      </tr>
      <?php
      $c = 1;
      while ($verificar_temperatura =
mysql_fetch_array($ejecutar_temperatura)) {
        if ( ($c % 2) == 0) {
          echo "<tr class='datos_cuerpo1'>";
          echo "<td>";
          echo $verificar_temperatura['Fecha'];
          echo "</td>";
          echo "<td>";
          echo $verificar_temperatura['Hora'];
          echo "</td>";
          echo "<td>";
          echo $verificar_temperatura['Grados1C'];
          echo "</td>";
          echo "<td>";
          echo $verificar_temperatura['Grados2C'];
          echo "</td>";
          echo "</tr>";
        }else{
          echo "<tr class='datos_cuerpo2'>";
          echo "<td>";
          echo $verificar_temperatura['Fecha'];
          echo "</td>";
          echo "<td>";
          echo $verificar_temperatura['Hora'];
          echo "</td>";
          echo "<td>";
          echo $verificar_temperatura['Grados1C'];
          echo "</td>";

```

```

        echo "<td>";
        echo $verificar_temperatura['Grados2C'];
        echo "</td>";
        echo "</tr>";
    }
    $c++;
}
?>
</table>
</fieldset>
<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Script que se encuentra en la cabecera de la página -->
<fieldset>
    <legend><h3>Gráfica: </h3></legend>
    <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
</fieldset>
<!-- En este formulario se puede observar solo un botón submit el cual se encargará
de mandar información
a la página pdf_temperatura.php por el método post, esta página redirigida se
encargará de crear un PDF, la información
a enviar, es el contenido guardado en las variables en PHP descritas al principio y
de tipo hidden para poder
utilizarlas en el documento a redirigir -->
<form action="pdf_temperatura.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
    <input type="hidden" id="t" name="temperatura1_inicio" value = "<?php
echo $temperatura1_inicio; ?>"/>
    <input type="hidden" id="t" name="temperatura1_final" value = "<?php
echo $temperatura1_final; ?>"/>
    <input class="sobre" type="submit" value="Imprimir datos por
temperatura1"/>
</form>
<br/>
</div>
</section>
<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>
<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Busqueda por la temperatura del segundo sensor

```
<?php
//Se incluye el inicio de sesión y la conexión a la base de datos
include("sesion.php");
include("conexion.php");
//Todos los datos mostrados en el select en home.php, estas son las temperaturas que se
//encuentran en la base de datos, se van a guardar en variables PHP por el método post
//para poder utilizarlas en una busqueda por temperatura de la base de datos
$temperatura2_inicio = $_POST["temperatura2_inicio_slc"];
$temperatura2_final = $_POST["temperatura2_final_slc"];
//En la variable buscar_temperatura2 se va a guardar la consulta SQL donde va a mostrar los datos
guardados
//en la tabla gradosc que se encuentren entre las temperaturas del sensor 2 propuestas por el usuario
$buscar_temperatura2 = "SELECT * FROM gradosc WHERE Grados2C BETWEEN ".$temperatura2_inicio."
AND ".$temperatura2_final." ORDER BY Grados2C ASC";
//En estas variables se ejecutan el código SQL con conexión a la base de datos y se guarda en las variables
$ejecutar_temperatura = mysql_query($buscar_temperatura2, $con);
$ejecutar_temperatura1 = mysql_query($buscar_temperatura2, $con);
$ejecutar_temperatura2 = mysql_query($buscar_temperatura2, $con);
?>
<!-- Nuevo estandar para HTML5 -->
<!DOCTYPE html>
<!-- Se inicializa HTML5 y se le agrega el idioma en el que se va a trabajar-->
<html lang="es">
<!-- En la cabecera se encontrará información como enlaces a otros archivos como CSS, tipo de acentuaciones
de las palabras,
el título de la página y Scrips para la creación de graficos interactivos, esta información el usuario no lo podra
observar
al momento de navegar sobre la página WEB -->
<head>
    <!-- Codificación universal, todos los caracteres no universales se van a poder interpretar en el
navegador -->
    <meta charset="utf-8" />
    <!-- Título de la página -->
    <title>Página principal</title>
    <!-- Se hace referencia al archivo con código CSS para ésta página WEB -->
    <link rel="stylesheet" type="text/css" href="css/misestilos.css">
    <!-- Se importan las librerias jquery -->
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
    <style type="text/css">
    ${demo.css}
    </style>
    <!-- Se define la función que creará la gráfica -->
    <script type="text/javascript">
    $(function () {
        $('#container').highcharts({
            //Se le da un título a la gráfica
            title: {
                text: <?php echo $temperatura2_inicio; ?> + ' °C a ' + <?php echo
$temperatura2_final; + ' °C' ?> ,
                x: -20 //center
            },
            //Se le puede agregar un subtítulo
            subtitle: {
                text: "",
                x: -20
            },
            //Se le agregan los datos interactivos en el eje de la x, se crea por medio de una búsqueda y
            //conexión a la base da datos
```

```

xAxis: {
  categories: [
    <?php
    $cont=0;
    $total = mysql_num_rows($ejecutar_temperatura1);
    while ($ver_hora = mysql_fetch_array($ejecutar_temperatura1)) {
      echo "".$ver_hora['Hora']."";
      $cont++;
      if($total!=$cont) echo ",";
    }
    <?php } ?>
  ],
  //Se le agrega un título al eje de la Y
  yAxis: {
    title: {
      text: 'Temperatura (°C)'
    },
    plotLines: [{
      value: 0,
      width: 1,
      color: '#808080'
    }]
  },
  //Se le da formato a la "tabla" que sale al pasar el mouse por los nodos
  tooltip: {
    valueSuffix: '°C'
  },
  legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle',
    borderWidth: 0
  },
  //Se le da nombre al tipo de gráfica a mostrar y se le agregan los datos interactivos en el eje de la Y,
  //se crea por medio de una búsqueda y conexión a la base da datos
  series: [{
    name: 'Temperatura2',
    data: [
      <?php
      $cont = 0;
      $total = mysql_num_rows($ejecutar_temperatura2);
      while ($ver_temperatura = mysql_fetch_array($ejecutar_temperatura2)) {
        $temp = $ver_temperatura['Grados2C'];
        echo $temp;
        $cont++;
        if($total != $cont){
          echo ",";
        }
      }
      <?php } ?>
    ]
  }
  });
</script>
<!-- Se importan las librerías de la gráfica y de la exportación -->
<script src="highcharts/js/highcharts.js"></script>
<script src="highcharts/js/modules/exporting.js"></script>
</head>
<!-- Cuerpo de la página WEB, en esta sección se mostrará todo lo que se vaya a visualizar en el navegador -->

```

```

<body>
  <!-- Encabezado o cabecera del cuerpo -->
  <header>
    <!-- El DIV define una división o una sección de un documento en HTML -->
    <div>
      <!-- Se agrega un título y a sus lados unas imagenes que trabajan como logotipos
de la página WEB -->
      <h1> Reportes De Temperatura </h1>
    </div>
  </header>
  <!-- Se agrega el menú o la barra de navegación, contiene una clase para ser usada en la página con
código CSS
y así poder darle estilo a la barra de navegación-->
  <nav class="nav_home">
    <!-- Se le agrega el nombre del usuario logueado por el usuario, contiene una clase para ser
usada en la página con código CSS
y así poder darle estilo al nombre del usuario logueado -->
    <p class="sesion"> "Usted está ingresando como <?php echo $_SESSION["usuario"]; ?>"
  </p>
  </nav>
  <!-- Se utiliza para generar secciones en la página WEB, aquí va toda la información presentada en la
página WEB -->
  <section class="section_home">

    <!-- Dentro del siguiente DIV se crea un ASIDE el cual se encarga de seccionar el contenido
aparte de
la información central, se crea un segmento lateral que contienen links a otras páginas
dentro de la misma
página WEB -->
    <div class="aside">
      <h2 class="centrar">Menu</h2>
      <aside class="espacio">
        <br/>
        <a href="password.php"><h3>Cambiar contraseña</h3></a>
        <a href="home.php"><h3>Reportes de temperatura</h3></a>
        <a href="contacto.php"><h3>Contacto</h3></a>
        <form class="centrar">
          <a href="salir.php"><input class="sobre" type="button"
name="salir_btn" value="Salir" /></a>
        </form>
      </aside>
    </div>
    <!-- Dentro de este DIV va contenida toda la información visible y central de la página WEB -
->
    <div class="informacion">
      <!-- Se le agrega un título central -->
      <h2 class="centrar">Datos de la temperatura <?php echo $temperatura2_inicio; ?>
°C a <?php echo $temperatura2_final; ?> °C</h2>
      <!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestran los
datos generados entre el
rango de temperaturas que se encuentran en la base de datos descritas por el
usuario , con sus fechas, horas y
temperaturas, todo dentro de tablas -->
      <fieldset>
        <legend><h3>Mostrar datos: </h3></legend>
        <table class="datos">
          <tr class="datos_cabecera">
            <td>Fecha </td>
            <td>Hora </td>
            <td>Temperatura1 °C</td>
            <td>Temperatura2 °C</td>

```

```

        </tr>
    <?php
    $c = 1;
    while ($verificar_temperatura =
mysql_fetch_array($ejecutar_temperatura)) {
        if ( ($c % 2) == 0) {
            echo "<tr class='datos_cuerpo1'>";
            echo "<td>";
            echo $verificar_temperatura['Fecha'];
            echo "</td>";
            echo "<td>";
            echo $verificar_temperatura['Hora'];
            echo "</td>";
            echo "<td>";
            echo $verificar_temperatura['Grados1C'];
            echo "</td>";
            echo "<td>";
            echo $verificar_temperatura['Grados2C'];
            echo "</td>";
            echo "</tr>";
        }else{
            echo "<tr class='datos_cuerpo2'>";
            echo "<td>";
            echo $verificar_temperatura['Fecha'];
            echo "</td>";
            echo "<td>";
            echo $verificar_temperatura['Hora'];
            echo "</td>";
            echo "<td>";
            echo $verificar_temperatura['Grados1C'];
            echo "</td>";
            echo "<td>";
            echo $verificar_temperatura['Grados2C'];
            echo "</td>";
            echo "</tr>";
        }
        $c++;
    }
    ?>
</table>
</fieldset>
<!-- Dentro del FIELDSET el cual se encarga de organizar grupos, se muestra la
gráfica resalizada por el
Script que se encuentra en la cabecera de la página -->
<fieldset>
    <legend><h3>Gráfica: </h3></legend>
    <div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
</fieldset>
<!-- En este formulario se puede observar solo un botón submit el cual se encargará
de mandar información
a la página pdf_temperatura2.php por el método post, esta página redirigida se
encargará de crear un PDF, la información
a enviar, es el contenido guardado en las variables en PHP descritas al principio y
de tipo hidden para poder
utilizarlas en el documento a redirigir -->
<form action="pdf_temperatura2.php" method="post" target="_blank"
enctype="application/x-www-form-urlencoded">
    <input type="hidden" id="t" name="temperatura2_inicio" value = "<?php
echo $temperatura2_inicio; ?>"/>
    <input type="hidden" id="t" name="temperatura2_final" value = "<?php
echo $temperatura2_final; ?>"/>

```

```

        <input class="sobre" type="submit" value="Imprimir datos por
temperatura2"/>
        </form>
        <br/>
    </div>
</section>
<!-- A este DIV se le agrega una clase la cual nos va a servir para ponerle estilos a la página WEB -->
<div class="barra_superior">
</div>
<!-- En esta sección se agrega en pie de la página WEB -->
<footer class="footer_home">
    <p>Universidad Autónoma del Carmen y Universidad de Quintana Roo</p>
</footer>
</body>
</html>

```

Respaldo en formato PDF con datos en general

```

<?php
//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");
//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{
    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){
        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(180, 10, "Datos generales", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 163, 10, 25, 25);
        $this -> Ln(30);
    }
}
//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura1", "Temperatura2"
);
//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();
//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();
//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {
    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(45, 10, $cabeceraT[$i], 1, 0, 'C', true);
}
//Se crea un salto de linea y se guarda en una variable
$mipdf -> Ln(10);
//En la variable mostrar_datos_general se va a guardar la consulta SQL donde va a mostrar los datos
guardados
//en la tabla gradosc
$mostrar_dato_general = "SELECT * FROM gradosc";
//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_dato_general = mysql_query($mostrar_dato_general, $con);
//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_dato_general = mysql_fetch_array($ejecutar_dato_general)) {

```

```

//Se mandan a llamar todos los datos de la tabla y se guarda en una variable para posteriormente
usarla
$fecha = $verificar_dato_general['Fecha'];
$hora = $verificar_dato_general['Hora'];
$temperatura1 = $verificar_dato_general['Grados1C'];
$temperatura2 = $verificar_dato_general['Grados2C'];
//Todos los datos se muestran dentro de una tabla con el color puesto por el programador
$mipdf -> SetFillColor(100, 100, 200);
$mipdf -> Cell(45, 10, $fecha, 1, 0, 'C', true);
$mipdf -> Cell(45, 10, $hora, 1, 0, 'C', true);
$mipdf -> Cell(45, 10, $temperatura1, 1, 0, 'C', true);
$mipdf -> Cell(45, 10, $temperatura2, 1, 0, 'C', true);
$mipdf -> Ln(10);
}
//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Respaldo en formato PDF por fecha

```

<?php
//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");
//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por fecha de la base de datos
$fecha_inicio = $_POST['fecha_inicio'];
$fecha_final = $_POST['fecha_final'];
//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{
    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){
        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(180, 10, "Datos ordenados por fecha", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 163, 10, 25, 25);
        $this -> Ln(30);
    }
}
//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura1", "Temperatura2"
);
//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();
//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();
//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {
    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(45, 10, $cabeceraT[$i], 1, 0, 'C', true);
}
//Se crea un salto de linea y se guarda en una variable
$mipdf -> Ln(10);
//En la variable mostrarr_fecha se va a guardar la consulta SQL donde va a mostrar los datos guardados

```

```

//en la tabla gradosc que se encuentren entre las fechas propuestas por el usuario
$mostrar_fecha = "SELECT * FROM gradosc WHERE Fecha BETWEEN '" . $fecha_inicio . "' AND
'" . $fecha_final . "' ";
//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_fecha = mysql_query($mostrar_fecha, $con);
//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_fecha = mysql_fetch_array($ejecutar_fecha)) {
    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
        $fecha = $verificar_fecha['Fecha'];
        $hora = $verificar_fecha['Hora'];
        $temperatura1 = $verificar_fecha['Grados1C'];
        $temperatura2 = $verificar_fecha['Grados2C'];
        //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
        $mipdf -> SetFillColor(100, 100, 200);
        $mipdf -> Cell(45, 10, $fecha, 1, 0, 'C', true);
        $mipdf -> Cell(45, 10, $hora, 1, 0, 'C', true);
        $mipdf -> Cell(45, 10, $temperatura1, 1, 0, 'C', true);
        $mipdf -> Cell(45, 10, $temperatura2, 1, 0, 'C', true);
        $mipdf -> Ln(10);
    }
//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Respaldo del formato en PDF por hora

```

<?php
//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");
//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por hora de la base de datos
$hora_inicio = $_POST['hora_inicio'];
$hora_final = $_POST['hora_final'];
//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{
    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){
        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont ('Arial', 'B', 20);
        $this -> Cell (180, 10, "Datos ordenados por hora", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 163, 10, 25, 25);
        $this -> Ln (30);
    }
}
//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura1", "Temperatura2"
);
//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();
//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();
//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {

```

```

    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(45, 10, $cabeceraT[$i], 1, 0, 'C', true);
}
//Se crea un salto de linea y se guarda en una variable
$mipdf -> Ln(10);
//En la variable mostrar_hora se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las horas propuestas por el usuario
$mostrar_hora = "SELECT * FROM gradosc WHERE Hora BETWEEN ".$hora_inicio." AND ".$hora_final." ";
//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_hora = mysql_query($mostrar_hora, $con);

//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_hora = mysql_fetch_array($ejecutar_hora)) {
    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
    $fecha = $verificar_hora['Fecha'];
    $hora = $verificar_hora['Hora'];
    $temperatura1 = $verificar_hora['Grados1C'];
    $temperatura2 = $verificar_hora['Grados2C'];

    //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
    $mipdf -> SetFillColor(100, 100, 200);
    $mipdf -> Cell(45, 10, $fecha, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $hora, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $temperatura1, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $temperatura2, 1, 0, 'C', true);
    $mipdf -> Ln(10);
}
//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Respaldo en formato PDF por la temperatura del primer sensor

```

<?php
//Se incluye la libreria fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");
//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por temperatura de la base de datos
$temperatura_inicio = $_POST['temperatura1_inicio'];
$temperatura_final = $_POST['temperatura1_final'];
//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{
    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imagenes y un título
    public function Header(){
        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(180, 10, "Datos ordenados por temperatura1", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 163, 10, 25, 25);
        $this -> Ln(30);
    }
}
//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura1", "Temperatura2"
);

```

```

//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();
//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();
//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {
    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(45, 10, $cabeceraT[$i], 1, 0, 'C', true);
}
//Se crea un salto de línea y se guarda en una variable
$mipdf -> Ln(10);
//En la variable mostrar_temperatura se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las temperaturas propuestas por el usuario
$mostrar_temperatura = "SELECT * FROM gradosc WHERE Grados1C BETWEEN ".$temperatura_inicio."
AND ".$temperatura_final." ";
//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_temperatura = mysql_query($mostrar_temperatura, $con);
//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_temperatura = mysql_fetch_array($ejecutar_temperatura)) {
    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
    $fecha = $verificar_temperatura['Fecha'];
    $hora = $verificar_temperatura['Hora'];
    $temperatura1 = $verificar_temperatura['Grados1C'];
    $temperatura2 = $verificar_temperatura['Grados2C'];

    //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
    $mipdf -> SetFillColor(100, 100, 200);
    $mipdf -> Cell(45, 10, $fecha, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $hora, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $temperatura1, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $temperatura2, 1, 0, 'C', true);
    $mipdf -> Ln(10);
}
//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Respaldo en formato PDF por la temperatura del segundo sensor

```

<?php
//Se incluye la librería fpdf y la conexión a la base de datos
include("conexion.php");
include("pdf/fpdf.php");
//Todos los datos enviados por el método oculto hidden
//se van a guardar en variables PHP por el método post
//para poder utilizarlas en una búsqueda por temperatura de la base de datos
$temperatura_inicio = $_POST['temperatura2_inicio'];
$temperatura_final = $_POST['temperatura2_final'];
//Se crea una clase, en esta clase se va a guardar todo lo que se va a visualizar en el PDF
class MiPDF extends FPDF{
    //se crea una función, esta función va a obtener el encabezado del PDF
    //Contiene dos imágenes y un título
    public function Header(){
        $this -> Image("pdf/logo_unacar.jpg", 10, 10, 25, 25);
        $this -> SetFont('Arial', 'B', 20);
        $this -> Cell(180, 10, "Datos ordenados por temperatura2", 0, 0, 'C');
        $this -> Image("pdf/logo_uqroo.jpg", 163, 10, 25, 25);
    }
}

```

```

        $this -> Ln (30);
    }
}
//Se crea un arreglo, este contendra una cabecera de una tabla que contendra el nombre
//de los datos que se van a mandar a llamar
$cabeceraT = array(
    "Fecha", "Hora", "Temperatura1", "Temperatura2"
);
//Se crea el PDF junto con su header y se guarda en una variable
$mipdf = new MiPDF();
//Se agrega una página en blanco y se guarda en una variable
$mipdf -> addPage();
//Se crea una tabla para las cabeceras que se crearon en el arreglo
//junto con el tipo de letra y color de texto
for ($i=0; $i < count($cabeceraT); $i++) {
    $mipdf -> SetFont('Courier', 'B', 15);
    $mipdf -> SetTextColor(255, 255, 255);
    $mipdf -> Cell(45, 10, $cabeceraT[$i], 1, 0, 'C', true);
}
//Se crea un salto de linea y se guarda en una variable
$mipdf -> Ln(10);
//En la variable mostrar_temperatura se va a guardar la consulta SQL donde va a mostrar los datos guardados
//en la tabla gradosc que se encuentren entre las temperaturas propuestas por el usuario
$mostrar_temperatura = "SELECT * FROM gradosc WHERE Grados2C BETWEEN ".$temperatura_inicio."
AND ".$temperatura_final." ";
//En esta variable se ejecuta el código SQL con conexión a la base de datos y se guarda en la variable
$ejecutar_temperatura = mysql_query($mostrar_temperatura, $con);
//Se recorren todos los datos de la tabla ejecutada con el código SQL
while ($verificar_temperatura = mysql_fetch_array($ejecutar_temperatura)) {
    //Se mandan a llamar todos los datos de la tabla y se guardan en una variable para posteriormente
    usarla
    $fecha = $verificar_temperatura['Fecha'];
    $hora = $verificar_temperatura['Hora'];
    $temperatura1 = $verificar_temperatura['Grados1C'];
    $temperatura2 = $verificar_temperatura['Grados2C'];
    //Todos los datos se muestran dentro de una tabla con el color puesto por el programador
    $mipdf -> SetFillColor(100, 100, 200);
    $mipdf -> Cell(45, 10, $fecha, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $hora, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $temperatura1, 1, 0, 'C', true);
    $mipdf -> Cell(45, 10, $temperatura2, 1, 0, 'C', true);
    $mipdf -> Ln(10);
}
//Se imprimen todos los datos
$mipdf -> Output();
?>

```

Datasheet sensor de temperatura DS18B20

Para saber más sobre las características y cómo funciona el sensor de temperatura sumergible DS18B20 para Arduino, hacer click en [DS18B20](#) para trasladarse a su datasheet.