



Universidad Autónoma de Campeche
Facultad de Ingeniería
Ingeniería en Sistemas Computacionales

Sistemas II

7º semestre, grupo "A"

Documentación del Proyecto de Programación Web

- Aguilar Rodríguez José F.
- Arguez Montejo Derian A.
- Aubry Ramírez Eduardo D.
- Campos Pérez Jonathan
- Herrera Palí Héctor A.
- Manzanilla Yeh Edgar C.
- Officer Segovia Isaac N.
- Trujillo Medina Daniel A.

28 de noviembre del 2018

REQUERIMIENTOS

CALC_CARGA_TERM

- Interfaz que pida los datos de la empresa.
- Características del cuarto en donde se colocará el clima.
- Sección que pida escoger el estado y municipio de la empresa.
- Cálculo de carga térmica donde se pide describir las dimensiones del cuarto, posiciones y dimensiones de las ventanas.
- Poder seleccionar el tipo de techo que se está usando.
- Una vez calculado todo, se mostrará al usuario una tabla con las especificaciones de los mejores climas que podrá utilizar en su habitación.

CALC CONS. ELECT

Datos de placa

- Seleccionar el tipo de equipo.

Unidad condensadora

- Pedir marca, modelo y su capacidad de enfriamiento.
- Pedir potencia activa (si en caso de que la unidad condensadora no venga con potencia activa se pedirá la información de la potencia activa por tensión y corriente).
- Pedir tensión, corriente normal, refrigerante y EER por datos de placa.

Unidad evaporadora/ manejadora de aire

- Pedir marca, modelo y capacidad de enfriamiento.

Mantenimiento

- Pedir la antigüedad del equipo, condiciones actuales del equipo, ciclo de operación y los días de operación al año.

MEDICIONES PUNTUALES

Medición con amperímetro de potencia

- Ingresa los valores de la potencia (KW), en tres determinados horarios
- Cada horario tendrá tres pruebas de medición de potencia donde el usuario insertará sus pruebas de medición.

Medición con multímetro de gancho

- Ingresar los valores de la tensión en tres determinados horarios.
- Cada horario tendrá tres pruebas de medición de potencia donde el usuario insertará sus pruebas de medición.

MEDICIONES PUNTUALES

- Especificar las condiciones de aire de entrada con un “select” (para Temperatura de Bulbo seco y para humedad relativa).
- Especificar las condiciones de aire de inyección con un “select” (para Temperatura de Bulbo seco y para humedad relativa).
- Que el usuario inserte la velocidad de inyección.
- Que el usuario inserte los valores del largo y ancho para el área del ducto de salida del evaporador.

MODELO DE ARQUITECTURA DE LA APLICACIÓN

El modelo utilizado es de Sistemas de información del MHC-PMS con cuatro capas:

1

Interfaz de navegador Web

2

Insertar Datos

Mostrar datos

3

Servlet (Realización de Operaciones)

4

Gestor de Base de Datos

Componentes:

1- **Capa 1:**

- a. **Interfaz de navegador Web:** Es la interfaz donde el usuario va a interactuar con el HTML insertando datos en campos específicos para calcular los datos (BTU) que le indicarán qué aire acondicionado es el óptimo para su uso.

2- **Capa 2:**

- a. **Insertar Datos:** El usuario insertará los datos en los campos en donde algunos son obligatorios para que se realicen las operaciones de manera correcta y algunos campos son opcionales, las cuáles si no son rellenas, el sistema lo toma como valor 0. Los datos que se hayan puesto se envían al Servlet en donde se realizarán las operaciones necesarias para enviar los datos de regreso al usuario.
- b. **Mostrar Datos:** Aquí se muestran los resultados de las operaciones realizadas anteriormente con los datos ingresados al Servlet.

3- **Capa 3:**

- a. **Servlet:** Es el código Java que se ejecuta en el servidor web. Recibe las peticiones o datos del cliente (Browser), las procesan y devuelven a éste un resultado al cliente. El Servlet se comunicará por medio de una clase llamada conexión en el que será el intermediario con la base de datos en donde se guardarán los datos que hayan puesto en los campos "insertar datos" y se consultarán sugerencias sobre algunos aires acondicionados que mejor se adecue al usuario.

4- **Capa 4:**

- a. **Gestor de Base de Datos:** Guarda y envía información al Servlet.

LENGUAJES DE PROGRAMACIÓN A UTILIZAR DURANTE EL PROYECTO

- Java 11.0.1 es un lenguaje de programación y una plataforma informática orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo.
- JSP (Java Server Pages) es una tecnología orientada a crear páginas web con programación en Java. Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.
- El motor de las páginas JSP está basado en los servlets de Java -programas en Java destinados a ejecutarse en el servidor-, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los servlets.
- JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.

CONVENCIONES

Declaración de las variables: se declararon las variables conforme a la información que se requerían.

```
float apAncho, apFondo;  
float vdrAlto, vdrAncho, vdrFondo;  
float vlsoeAlto, vlsoeAncho;  
float vlsAlto, vlsAncho;  
float vloAlto, vloAncho;  
float vlnosAlto, vlnosAncho;  
float arvAlto, arvAncho;  
float apu, arpe, atparna;  
  
float tccAncho, tccFondo;  
float tcpAncho, tcpFondo;  
float tcpaAncho, tcpaFondo;  
float tnaAncho, tnaFondo;
```

Comentarios: se abreviaron las variables para poder utilizarlas de mejor manera y se documentó todo el significado para que alguien más pueda entenderlo.

```
/*  
declaracion de las variables se abreviaron  
ap=Área de piso  
vdr=volumen del resinto  
vlsoe= Ventana lado sur o este  
vls= Ventana lado suroeste  
vlo= ventana lado oeste  
vlnos= ventana lado noreste o sureste  
arv= area del resto de las ventanas  
apu= area de la pared utilizada en el punto 3  
arpe= area del resto de las paredes exteriores  
atparna= area de todas las paredes adyacentes a un recinto no acondicionado  
tcc=techo con construccion sin acondicionar  
tcp= techo con plafon no aislado  
tcpa= techo con plafon con aislamiento  
tna= techo no aislado  
npar= numerp de personas dentro del resinto  
leens= luz y equipo electrico en uso  
*/
```

Conexión a la base de datos: se crearon los métodos para poder conectar a la base de datos por medio del jdbc a MySQL.

```
public void Connectin() throws SQLException{
    //Conecta la base de datos
    conn = null;
    try {
        Class.forName(driver);
        //Nos conectamos a la base de datos
        conn = DriverManager.getConnection(url, user, password);
        if (conn != null) {
            System.out.println("Conexion establecida");
        }
    } catch (ClassNotFoundException e) {
        System.out.println("Error al conectar" + e);
    }
}
```

Manejo de condiciones de error: nos mostrará los errores que se puedan presentar en la base de datos, como pueden ser: errores en la hora de conectar a la base de datos, los datos no se encontraron, etc.

```
/*
 *Se usa los try para poder arrojar los errores en caso de que se
 *se encuentre un error en la base de datos
 */
try{
    st = conn.createStatement();
    ResultSet rs = st.executeQuery(sql);

    while (rs.next())
    {
        System.out.println(rs.getObject("capacidad"));
        System.out.println(rs.getObject("marca"));
        System.out.println(rs.getObject("modelo"));
        System.out.println(rs.getObject("voltaje"));
        System.out.println(rs.getObject("corriente"));
        System.out.println(rs.getObject("refrigerante"));
        System.out.println(rs.getObject("potencia_electrica"));
        System.out.println(rs.getObject("seer"));
    }
}
```

Seguridad de la base de datos: solamente se podrá acceder por medio de la contraseña, usuario y la dirección de la base de datos que se implementaron con la conexión.

```
public class BD{  
  
    private Connection conn;  
    private static final String driver = "com.mysql.jdbc.Driver";  
    private static final String user = "root";//usuario  
    private static final String password = "";//contraseña  
    private static final String url = "jdbc:mysql://localhost:3306/clima_web_bd";//lugar donde esta ubicada la BD
```

HERRAMIENTAS PARA LA CONSTRUCCIÓN A UTILIZAR DURANTE EL PROYECTO

Cada una de las etapas llevadas a cabo en el desarrollo de software, cuenta con sus distintas herramientas. Estas herramientas tienen como finalidad, disminuir el estrés y los tiempos de cada fase; para que, además mejorar los resultados obtenidos y dar mejores propuestas al cliente. Además de que, claro, el nivel de satisfacción con el uso de herramientas es mucho mayor que sin ellas. Por eso, a continuación, vamos a ver las mejores herramientas para el desarrollo de software en cada una de las etapas o fases que conlleva el proceso.

En este proyecto usamos la herramienta de diagramas UML para presentar los componentes de nuestro sistema y su funcionamiento a desarrollar.

Una vez que ya contamos con lo que son los Diagramas UML en la parte del modelado de software, es momento de pasar al desarrollo. Esta etapa es de las más laboriosas y complicadas, dependiendo de tu equipo de trabajo, pero también del lenguaje de programación que estés dispuesto a utilizar; para la mayoría de los lenguajes conocidos, contamos con sus respectivos IDE.

Herramientas usadas durante la construcción de software:

1. **NetBeans 8.2:** es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.
Suele dar soporte a casi todas las novedades en el lenguaje Java. Cualquier preview del lenguaje es rápidamente soportada por Netbeans.
Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos frameworks.
Buen editor de código, multilenguaje, con el habitual coloreado y sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, plantillas de código, coding tips, herramientas de refactorización, y un largo etcétera. También hay tecnologías donde podemos usar el pulsar y arrastrar para incluir componentes en nuestro código.

Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario. Una vez que nos metemos en una clase java, por poner un ejemplo, se nos mostrarán distintas ventanas con el código, su localización en el proyecto, una lista de los métodos y propiedades (ordenadas alfabéticamente), también hay una vista que nos presenta las jerarquías que tiene nuestra clase y otras muchas opciones. Por supuesto personalizable según el gusto de cada usuario.

2. **JSP 2.1:** Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor. JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET desarrollado para la creación de aplicaciones web potentes. La tecnología de JSP permite a los desarrolladores y a los diseñadores de web desarrollar rápidamente y mantener fácilmente páginas dinámicas, ricas en información como son las que soportan a sistemas de negociación. La tecnología de los JSP separa la interfaz del usuario de la parte lógica del contenido permitiendo a los diseñadores cambiar a su disposición las plantillas de la interfaz sin alterar el contenido dinámico subyacente.

JSP también permite introducir código para la generación dinámica de HTML dentro de una página web. Esta surge por la necesidad de crear aplicaciones dinámicas para web de forma fácil, ya que la mejor parte del resultado de un programa CGI es estático. Se podría pensar entonces en JavaScript, pero este genera HTML dinámicamente en el cliente y no pueda entrar a los cursos del servidor.

Una de sus principales características de este lenguaje es que puede usarse el lenguaje Java en el servidor y el HTML en el navegador web, no necesariamente tiene que usarse HTML, si no también XML o WML. Como propósito final tiene la función de procesar una solicitud para crear una respuesta.

3. **JavaScript:** es uno de los más potentes e importantes lenguajes de programación en la actualidad, por tres enfoques claros: es útil, práctico y está disponible en cualquier navegador web. El poder de JavaScript está disponible principalmente en lado frontend, agregando mayor interactividad a la web, también puedes usar las librerías y framework como: jQuery, angular, Backbone, react y demás, escritas sobre JavaScript, y que te ayudan a crear una mejor experiencia de usuario en nuestros sitios web. De igual manera JavaScript se puede utilizar en los servidores web. Node.JS es tu mejor opción para usar este lenguaje del lado del servidor.

PROCEDIMIENTO DE INTEGRACIÓN

SCRUM marco de desarrollos ágiles.

Product Owner: Documento Excel, nos otorga la información para la realización del proyecto.

Equipo: equipo integrado que tiene la responsabilidad de entregar los avances del trabajo conformados con equipos de 2 personas encargadas de distintas tareas.

Scrum Máster: Daniel, encargado de identificar las peticiones del Excel y darles a los equipos sus trabajos requeridos.

Proceso

Equipos de 2 personas:

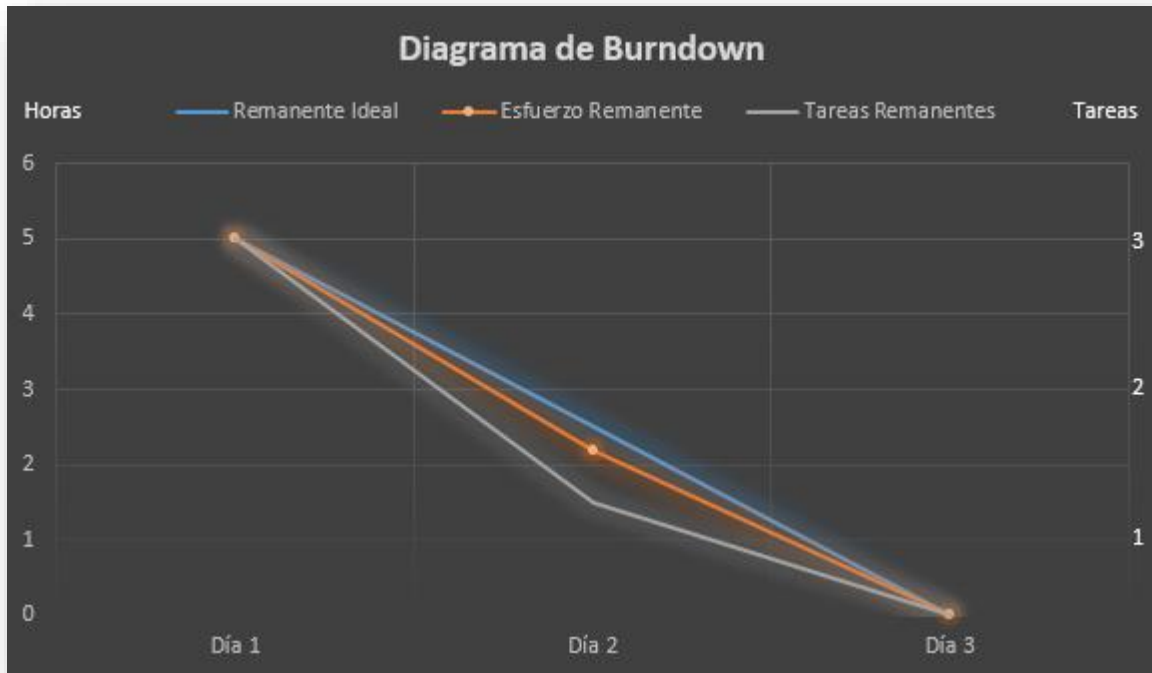
- Equipo 1: encargados de las bases de datos.
- Equipo 2: encargados del servlet.
- Equipo 3: encargados del html.

Los equipos quedan estables durante las dos semanas del proyecto.

Ciclos (Sprints 1):

- Duración:
 - Fueron dos días los acordados para la realización de los trabajos repartidos con los integrantes del equipo en base al documento Excel.
- Planificación:
 - Los equipos se reunirán en un lugar específico para el repartimiento de trabajo, especificando y analizando todas las partes necesarias del documento Excel.
 - Cada equipo se le fue otorgada la tarea que necesita el Excel para su funcionamiento, siendo dos equipos de dos personas; un equipo para la base de datos, uno para la interfaz y otro para el controlador.
- Stand-ups:
 - 5 minutos antes de empezar la plática para después llegar todos a la misma conclusión.

- Diagrama de Burn-down:



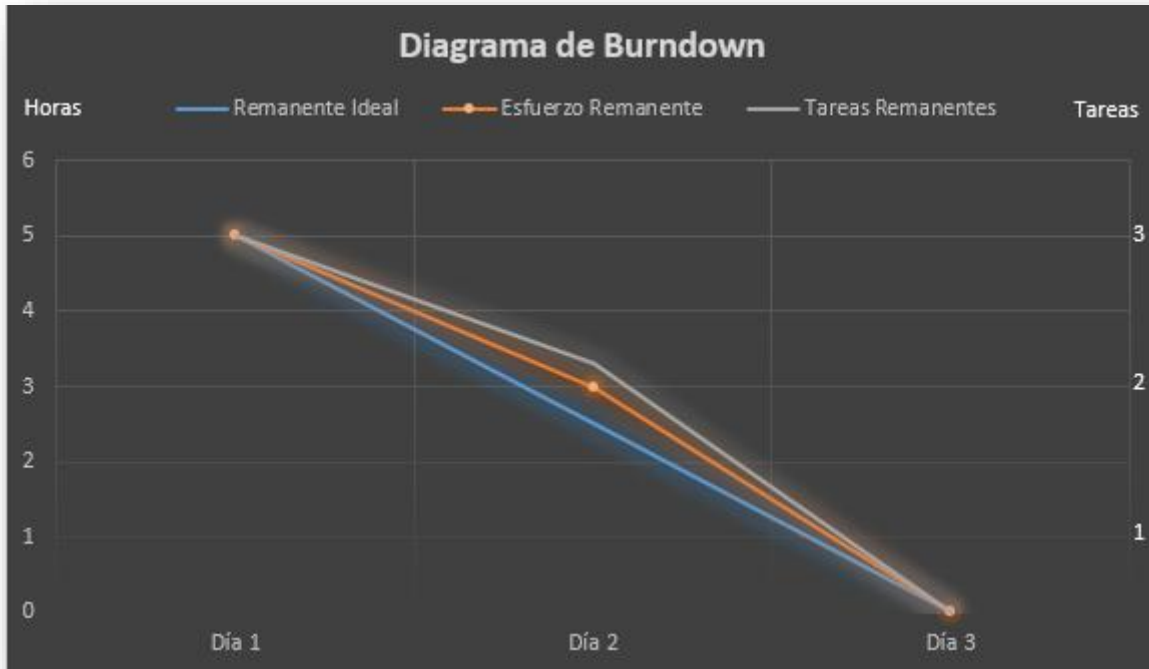
- Revisión:
 - El equipo demuestra que ha aprendido:
 - Analizar el Excel.
 - Idear la interfaz.
 - Idear los cálculos.
 - Idear la base de datos.
 - Especificar un controlador.
- Retrospectiva:
 - Se logro aprender sobre el Excel y el cómo funciona.
 - Se quedo en un acuerdo en el cómo se trabajar y se construirá el programa.

Ciclos (sprint2):

- Duración:
 - Los siguientes dos días se concentraron en la creación de la base de datos, analizando y observando el Excel.
- Planificación:
 - El equipo concentrado de la base de datos nos mostró el diagrama de Entidad/Relación.

- Con lo demás equipos se llegó a la conclusión de cambiar la base de datos en algunos aspectos.
- Se cambiaron los puntos a modificar de la base de datos.
- Stand-ups: 5 minutos antes de empezar la plática para después llegar todos a la misma conclusión.

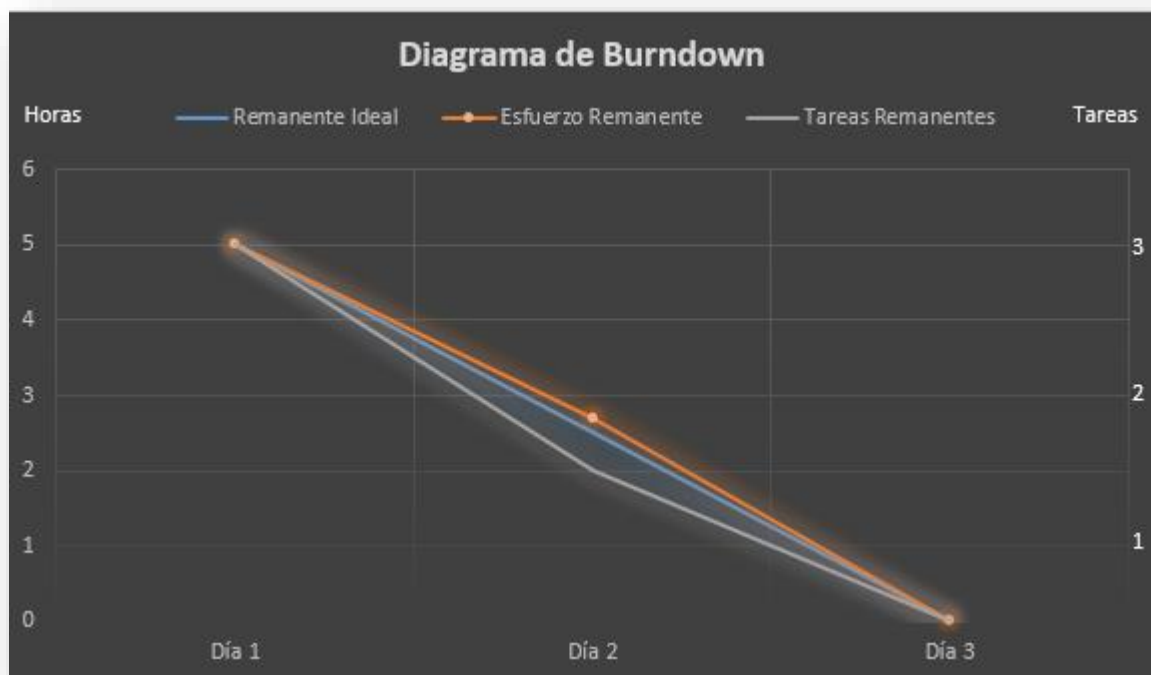
▪ Diagrama de Burn-down:



- Revisión:
 - El equipo demuestra que ha aprendido:
 - Relacionar las tablas de la base de datos basándonos de las hojas que tienen relación en el Excel.
 - Conectar un JSP en un Servlet.
 - Conectar una base de datos en un Servlet.
- Retrospectiva:
 - Se logro conectar la base de datos con el Servlet.
 - Se logro hacer consultas de la base de datos en el Servlet.

Ciclos (sprint3):

- Duración:
 - Los siguientes dos días se concentraron en la creación de un controlador y una interfaz web satisfaciendo todos los requerimientos del usuario, analizando y observando el Excel.
 - Planificación:
 - Los equipos concentrados en el controlador y en la vista.
 - Se mostro una interfaz donde el usuario va a interactuar con el programa.
 - Se mostro el Servlet donde llegaran las variables del usuario y comenzaran con las operaciones.
 - Stand-ups: 5 minutos antes de empezar la plática para después llegar todos a la misma conclusión.
- Diagrama de Burn-down:



- Revisión:
 - El equipo demuestra que ha aprendido. Opciones:
 - Manejar Servlet con base de datos y consultar sobre ella.
 - Consultar la base de datos desde un Servlet.
- Retrospectiva:
 - Se logró manejar el Servlet con la base de datos y consultar sobre ella.
 - Se logró consultar la base de datos mostrándolo en un JSP.

PLAN DE PRUEBAS FUNCIONALES

Ingreso de datos al sistema

En este apartado se verificaba por medio de impresiones como línea de comandos, los datos que recibía el Servlet, para comparar que eran los adecuados y ver que los factores multiplicativos eran los correctos para el municipio, ya que cada lugar tiene diferente grado de temperatura y esto ocasiona un cambio en los factores multiplicativos.

Para ver que eran los correctos, al ingresar los datos se imprimían los factores multiplicativos y se verificaban con el Excel para verificar que fueran los correctos.

Cálculo de los BTU

La estrategia para sistematizar las pruebas del proyecto en el cálculo de los BTU fue revisar uno a uno los requerimientos de usuario y compararlos manualmente con los resultados obtenidos para poder percibir si los resultados obtenidos son correctos.

La forma manual en la que lo comparamos fue revisar los resultados parciales para que todos coincidieran con los resultados obtenidos en el servidor. Los resultados que nos arrojaba el servidor eran impresos en línea de comando y éstos eran comparados con el resultado del documento Excel. Para todas las pruebas se usó como base el municipio de Campeche del estado de Campeche, ya que, en primera fase el servidor y la página web solo funcionaban para él mismo.

Prueba de consulta

Esta prueba se hacía por medio de impresiones en línea de comandos, para poder llegar a esto, primero se hacía todo el cálculo de los BTU, con los factores multiplicativos correctos, y teniendo esto hacíamos una consulta a la base de datos y finalmente imprimíamos los resultados obtenidos, estos se comparaban con las tablas que el Excel nos proporcionaba y con esto verificábamos que los datos eran correctos.

INFORME DE EJECUCIÓN DE LAS PRUEBAS FUNCIONALES.

Ingreso de datos al sistema

El resultado esperado del municipio escogido era Campeche, el cual se imprimió en consola correctamente; después se prosiguió a comparar los factores multiplicativos con los esperados, ya que en estos varía la temperatura de cada estado.

Nombre de prueba	Resultado esperado	Resultado Obtenido
Área del piso	52	52
Volumen del recinto	20	20
Ventana lado sur o este	540	540
Ventana lado suroeste	920	920
Ventana lado este	1260	1260
Ventana lado noreste o sureste	700	700
Área del resto de las ventanas	220	220
Área de la pared utilizado en el punto 3	180	180
Área del resto de las paredes exteriores	148	148
Área de todas las paredes adyacentes a un recinto no acondicionado	68	68
Techo con construcción arriba sin acondicionar	52	52
Techo con plafón no aislado	140	140
Techado con plafón y 50 mm o más de aislamiento	44	44
Techo no aislado	236	236
Número de personas dentro del recinto	600	600
Luz y equipos electrónicos en uso.	3	3

Cálculo de los BTU

Se compararon los resultados de los factores multiplicativos con los esperados en el Excel, teniendo en cuenta que se tomó como estándar Campeche y los datos que trae por defecto el Excel.

Nombre de prueba	Resultado esperado	Resultado Obtenido
Área del piso	3493.152	3493.152
Volumen del recinto	3694.68	3694.68
Ventana lado sur ó este	0	0
Ventana lado suroeste	0	0
Ventana lado este	0	0
Ventana lado noreste o sureste	10885	10885
Área del resto de las ventanas	0	0
Área de la pared utilizado en el punto 3	4198.5	4198.5
Área del resto de las paredes exteriores	0	0
Área de todas las paredes adyacentes a un recinto no acondicionado	807.84	807.84
Techo con construcción arriba sin acondicionar	0	0
Techo con plafón no aislado	0	0
Techado con plafón y 50 mm o mas de aislamiento	2955.744	2955.744
Techo no aislado	0	0
Número de personas dentro del recinto	18000	18000
Luz y equipos electrónicos en uso.	6516	6516
Total	50550.916	50550.916

Prueba de consulta

Se compararon las impresiones de los climas en consola, con la información que contenía la tabla del Excel para ver que tuvieran la misma información, la prueba se hacía con el resultado de BTU calculado con los datos default del Excel y en el estado de Campeche, este daba como resultado 50550.916, por tanto, se tomó la tabla de BTU's mayores a 36000.

Marca	Modelo del clima esperado.	Numero de BTU del clima esperado.	Impresión de los climas en consola
York	DCP36NWB21S	36000	York DCP36NWB21S 36000
York	DHR36NCB21S	36000	York DHR36NCB21S 36000
Carrier	53CIC363A	36000	Carrier 53CIC363A 36000
Carrier	53CIQ363A	36000	Carrier 53CIQ363A 36000
LG	SW362HP	34000	LG SW362HP 34000

CÓDIGO FUENTE DEL PROYECTO

✓ Repositorio en GitHub:



<https://github.com/IsaacOff/ClimaWeb.git>