

MANUAL TECNICO

ANDREA ALVAREZ 2011113797

BD1 PRIMER SEMESTRE

Descripción:

Se ha desarrollado esta aplicación para el apoyo a una empresa que provee servicios de transporte a través de varias rutas, esta aplicación debe administrar la compra de boletos como también la reservación de los mismos y tiene alta flexibilidad al momento de elegir un destino aun cuando este no sea un fin o inicio del trayecto, como el servicios de bus en general. En este caso se puede denotar que el pago por el trayecto si cambia con respecto al punto donde se finaliza el trayecto. Para ello requerí el uso de base de datos postgree para la administración de sistema que administran como también una interfaz amigable desarrollado en framework de express y orientado a node.js como a jade.

Organización del código fuente:

Para este proyecto por haberlo trabajado en el framework de express ya viene indicado el js servidor con el nombre de app y para cada página visible es decir .jade se necesita un .js que se encarga de reenderizar la pagina y mandar información tal es el caso muy parecido a php, se crearon 2 documentos con extensión jade los cuales son rutas y viajes y 2 documentos con extensión js también llamados rutas y viajes en los reenderizadores se controlan los queries que nos devuelven las consultas que hemos realizado de esta forma hacer las debidas verificaciones ahora bien para el momento de querer registrar datos dentro de la base de datos es necesario enviar por medio de sockets la información hacia el servidor que es el que se encarga de realizar la consulta de ingreso hacia la base de datos.

Adelante se brinda un poco de código para verificar el uso de modulos

extends layout

block content

```
h1= title
p Bienvenido to #{title}
p Area de viajes
input(type="text" id="prueba" name="prueba" value=info)
br
label(id="lb1")
script.
  var text1 = document.getElementById("prueba").value;
  var vector=text1.split(",");
  var myTable="<table><tr><td style='width: 100px; color: black; text-align:
right;'> Nombre </td>";
  myTable+="<td style='width: 100px; color: black; text-align: right;'> Factura
</td>";
  myTable+="<td style='width: 100px; color: black; text-align: right;'> Viajes
</td>";
  myTable+="<td style='width: 100px; color: black; text-align: right;'> Precio
</td></tr>";
  for(var i=0;i<vector.length-1;i++){
    try{
      myTable+="<tr><td style='width: 100px; text-align: right;'>" + vector[i] + "
</td>";
      myTable+="<td style='width: 100px; text-align: right;'> " + vector[i+1] + "
</td>";
```

```
        myTable+="<td style='width: 100px; text-align: right;'> " + vector[i+2] + "  
</td>";  
        myTable+="<td style='width: 100px; text-align: right;'> " + vector[i+3] + "  
</td></tr>";  
        i=i+3;  
    }catch(e){}  
    myTable+="</table>";  
    var t = document.getElementById('lb1');
```

Descripción de Módulos y dependencias:

Los módulos que se instalaron para poder usar node.js fueron varios:

PG: modulo que permite la interconexión de node hacia o con la base de datos solamente indicándole contraseña, usuario y puerto.

Socket.io= modulo que permite hacer conexiones entre paquetes relacionados en el proyecto principalmente dentro del proyecto permite el envío de queries hacia la base de datos.

Express: (versión 3.4.8) Nos brinda el framework para poder utilizar jade node.js y también provee sockets con un documento .json escalable.

Jade: (versión 1.3.0): nos permite un marco muy parecido a html pero más simplificado para el ámbito cliente, proporcionado por express.

Npm: es el comando de node.js que nos permite levantar un servidor para poder montar nuestro documento con extensión .js (servidor).

Diagrama Entidad relacion

