

## Describe las tecnologías que permiten implementar el modelo DevOps:

Git lab: es una herramienta la cual se utiliza en el modelo DevOps el cual ayuda en el desarrollo de software debido a la creación de repositorios y versionamiento de los proyectos lo cual facilita a los desarrolladores estar en sincronía dentro del mismo proyecto con los demás desarrolladores.

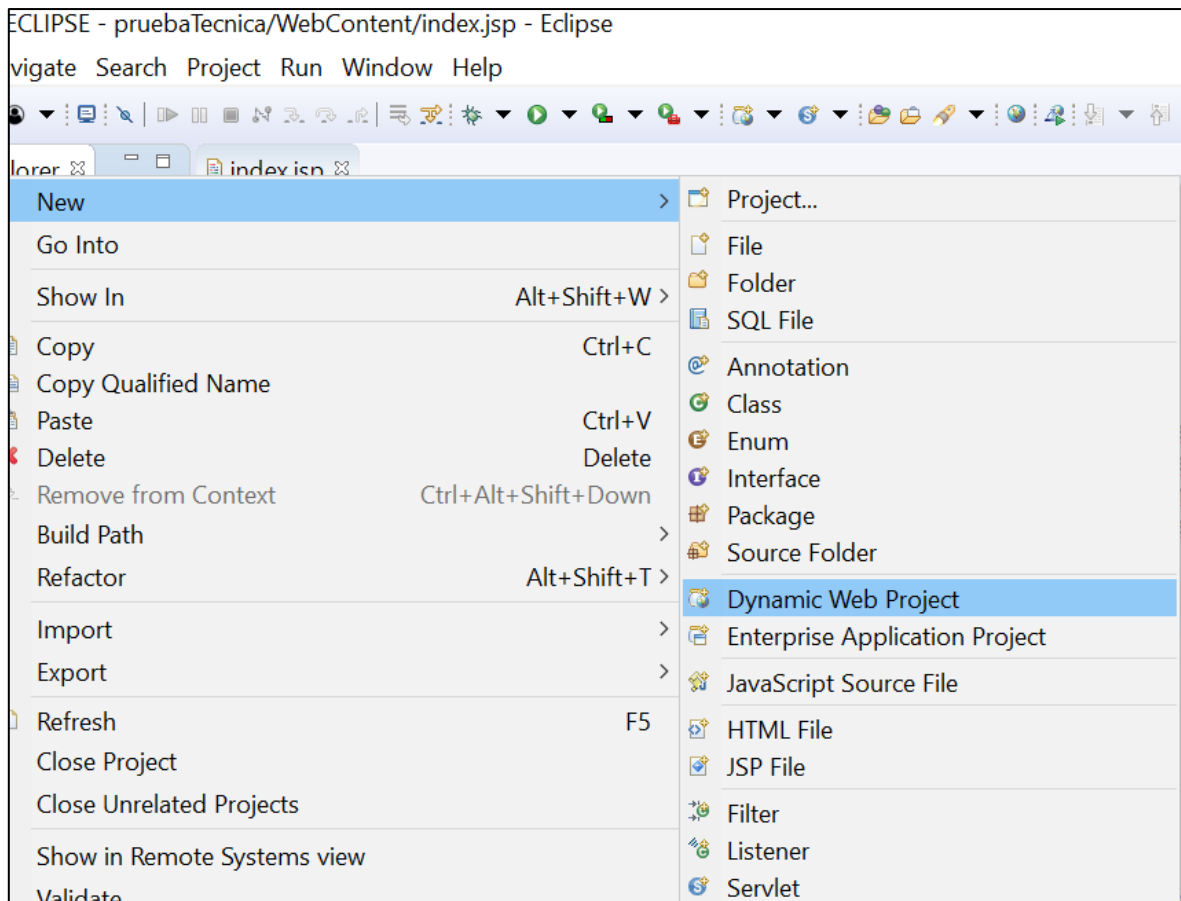
La metodología scrum se utiliza para el desarrollo de proyectos y ayuda a controlar tiempos y actividades en cada etapa del proyecto por lo que muy comúnmente se utiliza la herramienta trello para mantener este control de actividades

## Tecnologías utilizadas:

- **IDE:** Eclipse
- **Lenguaje:** Java 1.8
- **Proyecto:** Dynamic Web Project
- **Estructura:** MVC Servlets
- **Servidor de aplicaciones:** WildFly 10


## Réplica del sistema:


Creamos un nuevo proyecto en eclipse: **Dynamic Web Project**



Asignamos el nombre a nuestro proyecto.

## Dynamic Web Project



 A project already exists with this name.

Project name:

Project location

☒ Use default location

Location:

Target runtime

Dynamic web module version

Configuration

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership


☐ Add project to an EAR

EAR project name:


Working sets

☐ Add project to working sets

Working sets:



Damos clic en siguiente y seleccionamos la casilla de generar web.xml y damos clic en finalizar.

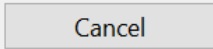
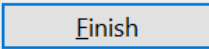
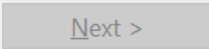
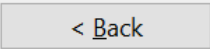

**Web Module**

Configure web module settings.

Context root:

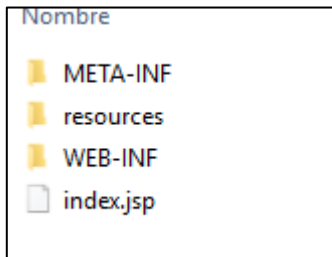
Content directory:

☒ Generate web.xml deployment descriptor



Nos creara la estructura de nuestro proyecto.

Dentro de la estructura WebContent agregamos nuestra primera vista index.jsp la cual por defecto reconoce nuestro proyecto al iniciar:



Creamos nuestra estructura, iniciando por un fieldset para que nuestros botones de accesos este dentro de un contenedor marcado.

Creamos 2 forms:

- 1- Llega al controlador IndexCO el cual redirige a la vista **main.jsp** para agregar una nueva organización
- 2- Llega al controlador IndexOrgCO el cual extrae la lista de organizaciones y la envía a la vista **orgList.jsp**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Prueba Tecnica</title>

<link rel="stylesheet" type="text/css" href="{pageContext.request.contextPath}/resources/css/bootstrap-3.3.7-dist/css/bootstrap.min.css"/>
<link rel="stylesheet" type="text/css" href="{pageContext.request.contextPath}/resources/css/utlis/common.css"/>
<script type="text/javascript" src="{pageContext.request.contextPath}/resources/js/jquery-3.3.1.js"></script>
<link rel="stylesheet" type="text/css" href="{pageContext.request.contextPath}/resources/css/index.css">
</head>
<body>

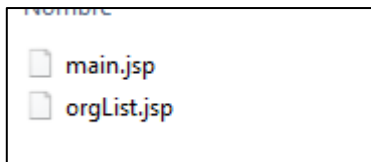
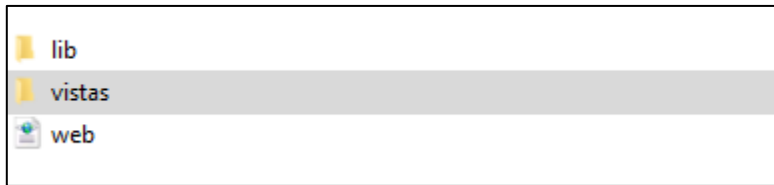
<br><br>
<br>
<br>
<br>

<fieldset style="width: 25%; margin: 0 auto">
  <form id="formIndex" name="nFormIndex"
    action="{pageContext.request.contextPath}/IndexCO" method="post"
    autocomplete="off">

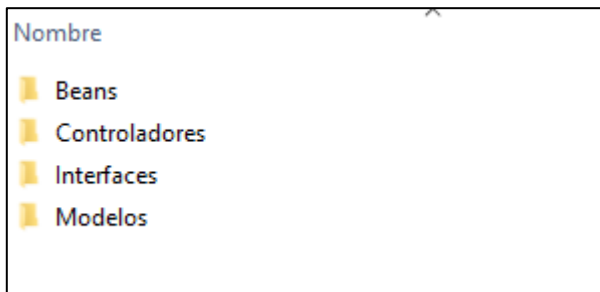
    <br>
    <table style="margin: 0 auto">
      <tr>
        <td><button id="submitBtn" class="btn btn-primary"
          type="submit">Nueva Organizacion</button></td>
      </tr>
    </table>
  </form>
  <form id="formIndex2" name="nFormIndex2"
    action="{pageContext.request.contextPath}/IndexOrgCO" method="post"
    autocomplete="off">

    <br>
    <table style="margin: 0 auto">
      <tr>
        <td><button id="submitBtn" class="btn btn-primary"
          type="submit">Lista de Organizaciones</button></td>
      </tr>
    </table>
  </form>
</fieldset>
```

Dentro del WEB-INF creamos una carpeta llamada vistas, la cual contendrá nuestras vistas.jsp



En la raíz de src creamos los siguientes paquetes:



Creamos el controlado IndexCo dentro del acarpeta de controladores el cual contiene lo siguiente:

Declaramos en la línea 19 que se trata de un servlet el cual por nombre contiene el mismo nombre del controlador.

En la línea 38 recibimos peticiones por método post y agregamos la vista al dispatcher en el request la cual se redirige a la vista en la línea 43

```

6  import java.util.ArrayList;
7
8  import javax.servlet.ServletException;
9  import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13
14
15
16 /**
17  * Servlet implementation class IndexCO
18  */
19 @WebServlet(name="IndexCO",urlPatterns={"/IndexCO"})
20 public class IndexCO extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     /**
24      * @see HttpServlet#HttpServlet()
25      */
26     public IndexCO() {
27         super();
28         // TODO Auto-generated constructor stub
29     }
30
31
32     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
33         // TODO Auto-generated method stub
34
35         request.getRequestDispatcher("index.jsp").forward(request, response);
36     }
37
38     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
39         // TODO Auto-generated method stub
40         request.setCharacterEncoding("UTF-8"); /** Problemas Tildes **/
41
42         request.getRequestDispatcher("/WEB-INF/vistas/main.jsp").forward(request, response);
43         return;
44     }
45 }

```

En la vista que creamos anteriormente main.jsp en la carpeta de vistas, agregamos la siguiente estructura:

- 1- tabla con los campos de la información de la organización
- 2- referenciamos en la línea 18 a nuestro archivo main.js el cual contiene la función que se llama en la línea 51 al llenar el campo de teléfono

```

18 <script type="text/javascript" src="${pageContext.request.contextPath}/resources/js/main.js"></script>
19 <body>
20
21 <form id="formIndex" name="nFormIndex" action="${pageContext.request.contextPath}/OrganizacionCO" method="post" >
22 <div class="wrapper">
23 <!-- Sidebar -->
24
25 <!-- Page Content -->
26 <div id="content">
27 <main role="main">
28 <hr>
29 <div class="container">
30
31 <div class="card">
32 <h4 class="card-header"><strong>Informacion de la organizacion</strong></h4>
33 <div class="card-body">
34
35 <hr>
36 <table class="table table-hover">
37 <thead class="thead-light">
38 <tr>
39 <th scope="col">Nombre</th>
40 <th scope="col">Direccion</th>
41 <th scope="col">Telefono</th>
42 <th scope="col">Id externo</th>
43 <th scope="col">codigo de encriptacion</th>
44 </tr>
45 </thead>
46 <tbody>
47 <tr>
48 <td><input type="text" id="strNombre" name="strNombre" required/></td>
49 <td><input type="text" id="strDireccion" name="strDireccion" required/></td>
50 <td><input type="text" id="strTelefono" name="strTelefono" onkeyup="this.value=Numeros(this.value)" Onchange="generaId();" minlength="10"
51 <td><input type="text" id="strIdExt" name="strIdExt" readonly/></td>
52 <td><input type="text" id="strCoEnc" name="strCoEnc" minlength="4" maxlength="4" required/></td>
53 </tr>
54 </tbody>
55 </table>
56 </div>
57 </div>
58 </div> <!-- /container -->
59
60 </main>
61 </div>
62
63 <table style="margin: 0 auto">
64 <tr>
65 <td><button id="submitBtn" class="btn btn-primary"
66 type="submit">Guardar</button></td>
67 </tr>

```

Creamos nuestro archivo main.js en la ruta WebContent/resources/

El cual contiene la siguiente función la cual extrae el valor de los campos según el id **strNombre** y **strTelefono** para extraer los primeros 4 dígitos del nombre y los últimos del teléfono y generar el Id Externo para ser asignado en la línea 13 al elemento con el id **strIdExt**.

```

5
6 function generaId() {
7
8     var id = null;
9     var nombre = document.getElementById("strNombre").value;
10    var telefono = document.getElementById("strTelefono").value;
11    var rand= Math.round(Math.random()*(999-001)+parseInt(001));
12    id=nombre.substr(-20,4)+telefono.slice(-4).toString()+"NE"+rand;
13    document.getElementById("strIdExt").value=id;
14
15 }
16

```

La vista main.jsp envía el contenido del formulario por método post al siguiente controlador:

### **OrganizacionCO**

En el cual declaramos nuestras variables y se asigna el valor recibido los parámetros enviados por la vista.

Creamos nuestro objeto de la clase Organización y asignamos las variables a los métodos set del objeto

Creamos nuestra instancia de la interfaz en la línea 52 para acceder al método de insert.

Al final redirigimos a la vista main.jsp

```

23 @WebServlet(name="OrganizacionCO",urlPatterns={"/OrganizacionCO"})
24 public class OrganizacionCO extends HttpServlet {
25     private static final long serialVersionUID = 1L;
26
27     /**
28      * @see HttpServlet#HttpServlet()
29      */
30     public OrganizacionCO() {
31         super();
32         // TODO Auto-generated constructor stub
33     }
34
35
36     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37         // TODO Auto-generated method stub
38
39         request.getRequestDispatcher("index.jsp").forward(request, response);
40     }
41
42     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
43         // TODO Auto-generated method stub
44         request.setCharacterEncoding("UTF-8"); /** Problemas Tildes **/
45
46         String strNombre = request.getParameter("strNombre");
47         String strDireccion = request.getParameter("strDireccion");
48         String strTelefono = request.getParameter("strTelefono");
49         String strIdExt = request.getParameter("strIdExt");
50         String strCoEnc = request.getParameter("strCoEnc");
51
52         IOrganizacion orgDao = new OrganizacionDAO();
53
54         Organizacion org= new Organizacion();
55         org.setStrNombre(strNombre);
56         org.setStrDireccion(strDireccion);
57         org.setStrTelefono(strTelefono);
58         org.setStrCoEnc(strCoEnc);
59         org.setStrIdExt(strIdExt);
60
61         try {
62             orgDao.insert(org);
63         } catch (Exception e) {
64             // TODO Auto-generated catch block
65             e.printStackTrace();
66         }
67
68         request.getRequestDispatcher("/WEB-INF/vistas/main.jsp").forward(request, response);
69         return;

```



Creamos la clase Bean Organización la cual contiene los atributos de la organización y sus métodos get & set en la carpeta de Beans

```
package Beans;

public class Organizacion {

    private String strNombre;
    private String strDireccion;
    private String strTelefono;
    private String strIdExt;
    private String strCoEnc;

    public String getStrNombre() {
        return strNombre;
    }
    public void setStrNombre(String strNombre) {
        this.strNombre = strNombre;
    }
    public String getStrDireccion() {
        return strDireccion;
    }
    public void setStrDireccion(String strDireccion) {
        this.strDireccion = strDireccion;
    }
    public String getStrTelefono() {
        return strTelefono;
    }
    public void setStrTelefono(String strTelefono) {
        this.strTelefono = strTelefono;
    }
    public String getStrIdExt() {
        return strIdExt;
    }
    public void setStrIdExt(String strIdExt) {
        this.strIdExt = strIdExt;
    }
    public String getStrCoEnc() {
        return strCoEnc;
    }
    public void setStrCoEnc(String strCoEnc) {
        this.strCoEnc = strCoEnc;
    }
}
```

Creamos la interfaz ***IOrganizacion*** en la carpeta de Interfaces:

```

1  package Interfaces;
2
3  import java.sql.SQLException;
4  import java.util.List;
5
6  import Beans.Organizacion;
7
8
9
10
11 public interface IOrganizacion {
12     public boolean insert(Organizacion org) throws Exception;
13     public List<Organizacion> findAll() throws Exception;
14 }
15

```

Creamos el modelo e implementamos la interfaz ***IOrganizacion*** con los métodos ***insert*** y ***findAll***

Método ***insert***:

```

9
10 import Beans.Organizacion;
11 import Interfaces.IOrganizacion;
12
13 public class OrganizacionDAO implements IOrganizacion {
14
15     @Override
16     public boolean insert(Organizacion org) throws Exception {
17
18         String strNombre= org.getStrNombre();
19         String strDireccion=org.getStrDireccion();
20         String strTelefono=org.getStrTelefono();
21         String strIdExt= org.getStrIdExt();
22         String strCoEnc=org.getStrCoEnc();
23         // TODO Auto-generated method stub
24
25         boolean bandera = false;
26         FileWriter flwriter = null;
27         try {
28             //crea el flujo para escribir en el archivo
29             flwriter = new FileWriter("C:\\Organizaciones\\organizaciones.properties");
30             //crea un buffer o flujo intermedio antes de escribir directamente en el archivo
31             BufferedWriter bfwriter = new BufferedWriter(flwriter);
32
33             //escribe los datos en el archivo
34             bfwriter.write(""+strNombre+" "+strDireccion+" "+strTelefono+" "+strIdExt+" "+strCoEnc);
35
36             //cierra el buffer intermedio
37             bfwriter.close();
38             bandera = true;
39         }
40         catch (Exception e){
41             System.out.println(e);
42             bandera = false;
43         }
44         System.out.println(bandera);
45         return bandera;
46     }
47

```

Método findAll:

```

50      @Override
51      public List<Organizacion> findAll() throws Exception {
52          List<Organizacion> listOrg= new ArrayList<Organizacion>();
53
54          try {
55              File file = new File("C:\\Organizaciones\\organizaciones.properties");
56              Scanner scanner = new Scanner(file, "UTF-8");
57
58
59              while (scanner.hasNextLine()) {
60                  String linea = scanner.nextLine();
61                  Scanner delimitar = new Scanner(linea);
62                  delimitar.useDelimiter("\\s*;\\s*");
63                  Organizacion org = new Organizacion();
64
65                  org.setStrNombre(delimitar.next());
66                  org.setStrDireccion(delimitar.next());
67                  org.setStrTelefono(delimitar.next());
68                  org.setStrIdExt (delimitar.next());
69                  org.setStrCoEnc(delimitar.next());
70                  listOrg.add(org);
71              }
72          } catch (Exception e) {
73
74          }
75          return listOrg;
76      }
77
78  }
79

```

Creamos controlador IndexOrgCO.java el cual realiza un llamado al método findAll para redirigir a la vista orgList.jsp con la información de las empresas.

```

23  @WebServlet(name="IndexOrgCO",urlPatterns={"/IndexOrgCO"})
24  public class IndexOrgCO extends HttpServlet {
25      private static final long serialVersionUID = 1L;
26
27      /**
28       * @see HttpServlet#HttpServlet()
29       */
30      public IndexOrgCO() {
31          super();
32          // TODO Auto-generated constructor stub
33      }
34
35
36      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37          // TODO Auto-generated method stub
38
39          request.getRequestDispatcher("index.jsp").forward(request, response);
40      }
41
42      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
43          // TODO Auto-generated method stub
44          request.setCharacterEncoding("UTF-8"); /** Problemas Tildes **/
45
46
47          IOrganizacion orgDao = new OrganizacionDAO();
48          List<Organizacion> orgList = new ArrayList<Organizacion>();
49          try {
50              List<Organizacion> orgList = orgDao.findAll();
51              if (null!=orgList&&orgList.size()>0) {
52                  request.setAttribute("orgList", orgList);
53              }
54          } catch (Exception e) {
55              // TODO Auto-generated catch block
56              e.printStackTrace();
57          }
58          request.getRequestDispatcher("/WEB-INF/vistas/orgList.jsp").forward(request, response);
59          return;
60      }
61

```

Creamos la estructura de nuestra vista orgList.jsp

Recibimos la lista con la información de la empresa enviada por el controlador y la recorremos con un `forEach` mostrando cada registro en la tabla.

Ocultamos la información de la organización y solo dejamos visible en nombre.

De la línea 46 a la 61 creamos el modal el cual será mostrado con la información de la empresa si se ingresa la clave correcta, de inicio está oculto.

```

21
22 <table id="TableRO" class="table table-striped table-bordered table-condensed">
23 <thead class="xxbstheadbg">
24 <tr>
25 <th>NOMBRE</th>
26 <th style="display:none;">DIRECCION&nbsp;&nbsp;&nbsp;</th>
27 <th style="display:none;">TELEFONO&nbsp;&nbsp;&nbsp;</th>
28 <th style="display:none;">ID&nbsp;&nbsp;&nbsp;EXTERNO&nbsp;&nbsp;&nbsp;</th>
29 <th style="display:none;">CODIGO&nbsp;&nbsp;&nbsp;ENC&nbsp;&nbsp;&nbsp;</th>
30
31 </tr></thead>
32 <tbody class="xxbstbodybg">
33 <c:forEach var="orgList" items="${orgList}" varStatus="row">
34 <tr>
35 <td id="nombre" class="boton">${orgList.strNombre}</td>
36 <td id="direccion" style="display:none;">${orgList.strDireccion}</td>
37 <td id="telefono" style="display:none;">${orgList.strTelefono}</td>
38 <td id="id" style="display:none;">${orgList.strIdExt}</td>
39 <td id="numero" style="display:none;">${orgList.strCoEnc}</td>
40
41 </tr>
42 </c:forEach>
43 </tbody>
44 </table>
45
46 <button id="btnModal" name="btnModal" style="display:none;">Abrir modal</button>
47 <div id="tvesModal" class="modalContainer">
48 <div class="modal-content">
49 <span class="close">x</span>
50 <h2>Informacion de Organizacion</h2>
51 <table>
52 <tr>
53 <td>Nombre: <input type="text" id="strNombre" name="strNombre" readonly/></td>
54 <td>Direccion: <input type="text" id="strDireccion" name="strDireccion" readonly/></td>
55 <td>Telefono: <input type="text" id="strTelefono" name="strTelefono" readonly/></td>
56 <td>Id Exterior<input type="text" id="strIdExt" name="strIdExt" readonly/></td>
57 <td>Codigo<input type="text" id="strCoEnc" name="strCoEnc" readonly/></td>
58 </tr>
59 </table>
60 </div>
61 </div>
62

```

Agregamos un id a cada columna de la tabla donde se imprimen los valores, pero además agregamos una clase botón al campo nombre para que cuando sea presionado se ejecute la siguiente función de js:

- 1- obtenemos los valores del renglón al cual se dio clic en la tabla y lo asignamos a una variable
- 2- mostramos una ventana en pantalla para que se ingrese la clave.
- 3- Validamos la clave ingresada con la del registro en la tabla.

```

65  $(document).ready(function() {
66      $("#ok").click(function() {
67          var clave = "";
68
69          $(this).parents("tr").find("#numero").each(function() {
70              clave += $(this).html() + "\n";
71          });
72
73          var textoEscrito = prompt("Ingresa la clave", "");
74          if(textoEscrito != clave){
75              alert("No has ingresado la clave correcta");
76          } else {
77              alert("la clave es correcta");
78
79              $("#btnModal").click();
80          }
81      });
82      $(".boton").click(function() {
83
84          // Obtenemos todos los valores contenidos en los <td> de la fila
85          // seleccionada
86          var clave = "";
87          $(this).parents("tr").find("#numero").each(function() {
88              clave += $(this).html() + "\n";
89          });
90
91          var nombre = "";
92          $(this).parents("tr").find("#nombre").each(function() {
93              nombre += $(this).html() + "\n";
94          });
95
96          var direccion = "";
97          $(this).parents("tr").find("#direccion").each(function() {
98              direccion += $(this).html() + "\n";
99          });
100
101          var telefono = "";
102          $(this).parents("tr").find("#telefono").each(function() {
103              telefono += $(this).html() + "\n";
104          });
105
106          var id = "";
107          $(this).parents("tr").find("#id").each(function() {
108              id += $(this).html() + "\n";
109          });
110

```

Si coincide se agregan los valores a la estructura de nuestro modal html y se simula un clic al botón oculto en nuestra vista el cual activa nuestro modal

```

var textoEscrito = prompt("Ingresa la clave", "");
if(textoEscrito != clave){
    alert("No has ingresado la clave correcta");
} else {
    alert("la clave es correcta");
    document.getElementById("strNombre").value=nombre;
    document.getElementById("strDireccion").value=direccion;
    document.getElementById("strTelefono").value=telefono;
    document.getElementById("strIdExt").value=id;
    document.getElementById("strCoEnc").value=clave;

    $("#btnModal").click();
}

});

```

La siguiente función detecta el clic del botón oculto en nuestra vista el cual muestra el modal con la información de la empresa.

```

32 if(document.getElementById("btnModal")){
33     var modal = document.getElementById("tvesModal");
34     var btn = document.getElementById("btnModal");
35     var span = document.getElementsByClassName("close")[0];
36     var body = document.getElementsByTagName("body")[0];
37
38
39     btn.onclick = function() {
40         modal.style.display = "block";
41
42         body.style.position = "static";
43         body.style.height = "100%";
44         body.style.overflow = "hidden";
45     }
46
47     span.onclick = function() {
48         modal.style.display = "none";
49
50         body.style.position = "inherit";
51         body.style.height = "auto";
52         body.style.overflow = "visible";
53     }
54
55     window.onclick = function(event) {
56         if (event.target == modal) {
57             modal.style.display = "none";
58
59             body.style.position = "inherit";
60             body.style.height = "auto";
61             body.style.overflow = "visible";
62         }
63     }
64 }

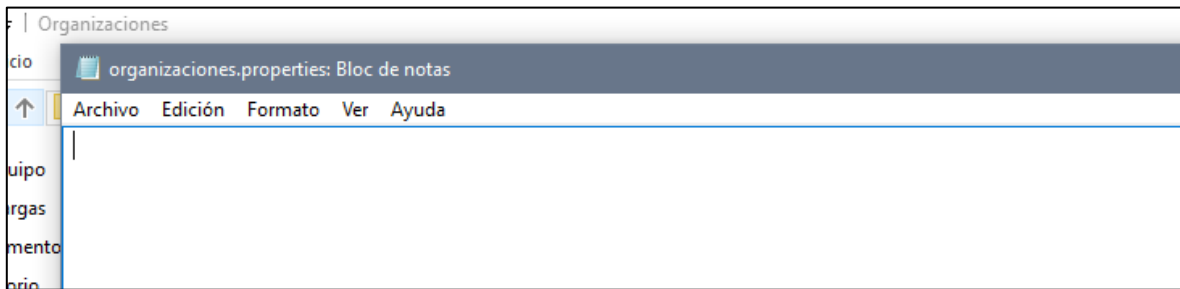
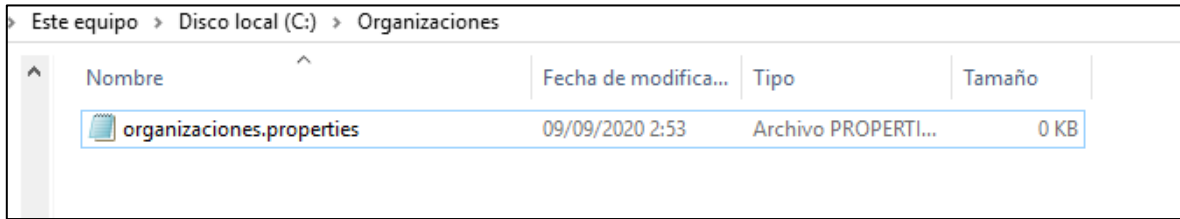
```

**Nota\*** Colocar la carpeta organizaciones en C: para correcta lectura de archivo o modificar la ruta que se encuentra en el archivo: OrganizacionesDAO en la línea 29 y 55

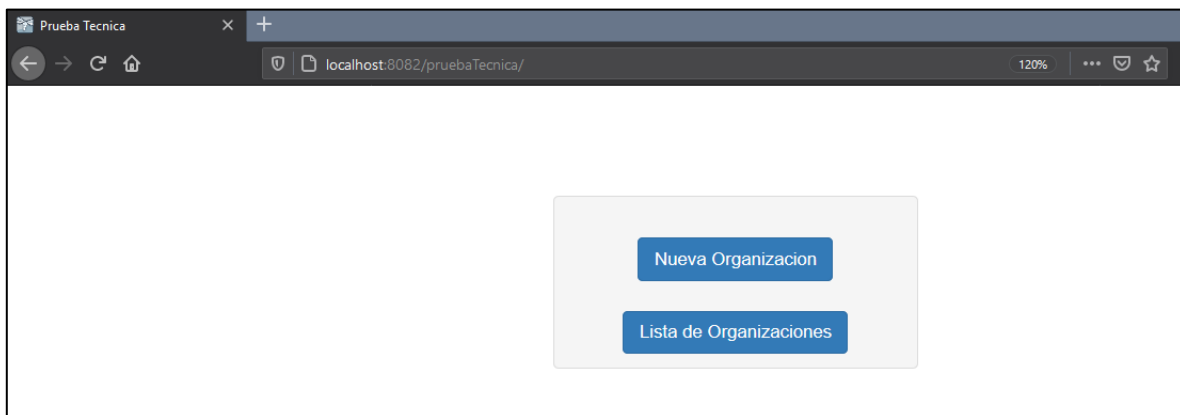


## Funcionalidad del sistema

Documento Organizaciones.properties en la ruta: **C:\\Organizaciones** se muestra vacío



Ingresamos a la página principal de nuestro proyecto



Para dar de alta una nueva organización presionamos el botón de Nueva Organización el cual nos redirige a la ventana para agregar una nueva.

Prueba Tecnica

localhost:8082/pruebaTecnica/IndexCO

### Informacion de la organizacion

Nombre	Direccion	Telefono	Id externo	codigo de encriptacion
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

[Guardar](#)

Agregamos los campos de la nueva organización, el campo Id externo es solo de lectura y se llena automáticamente agregando el nombre y el número de teléfono.

Al llenar todos los campos, damos clic en guardar.

Prueba Tecnica

localhost:8082/pruebaTecnica/IndexCO

### Informacion de la organizacion

Nombre	Direccion	Telefono	Id externo	codigo de encriptacion
Francisco	irazu 102	5545676543	Fran6543NE733	1234

[Guardar](#)

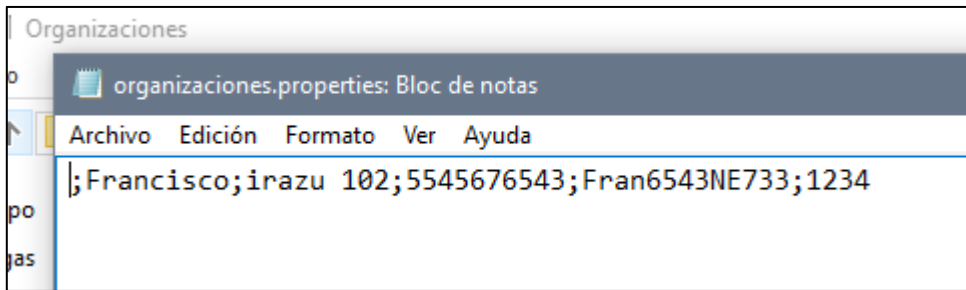
Podemos observar que el peso de nuestro archivo ha aumentado.

Este equipo > Disco local (C:) > Organizaciones

Nombre	Fecha de modifica...	Tipo	Tamaño
organizaciones.properties	09/09/2020 2:56	Archivo PROPERTI...	1 KB

Dentro contiene la información que guardamos anteriormente.





Agregamos un registro más...

Prueba Tecnica

localhost:8082/pruebaTecnica/OrganizacionCO

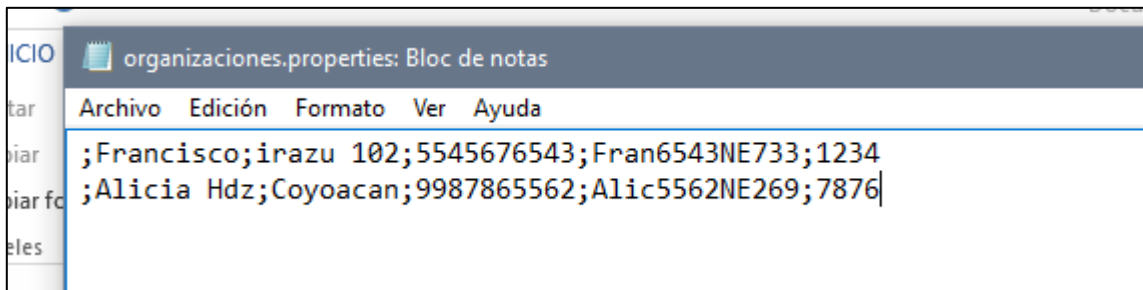
120%

**Información de la organización**

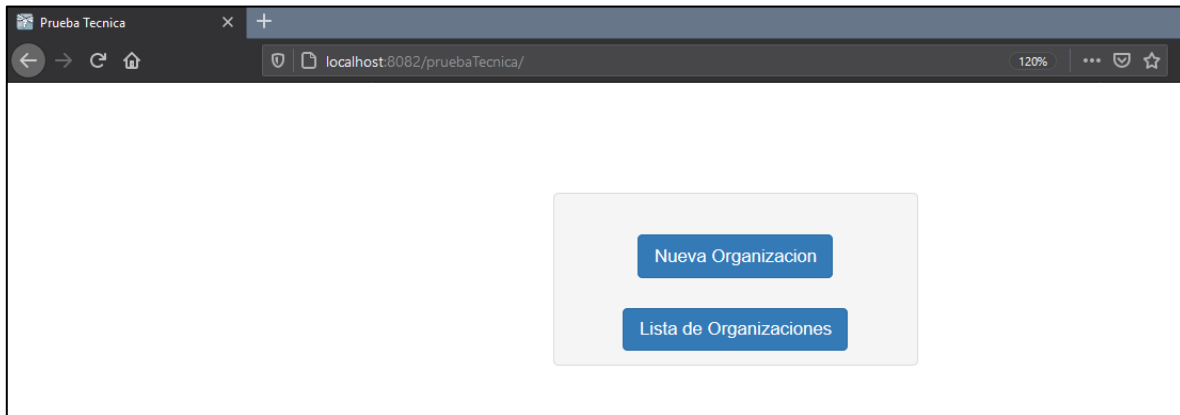
Nombre	Direccion	Telefono	Id externo	codigo de encriptacion
Alicia Hdz	Coyoacan	9987865562	Alic5562NE269	7876

Guardar

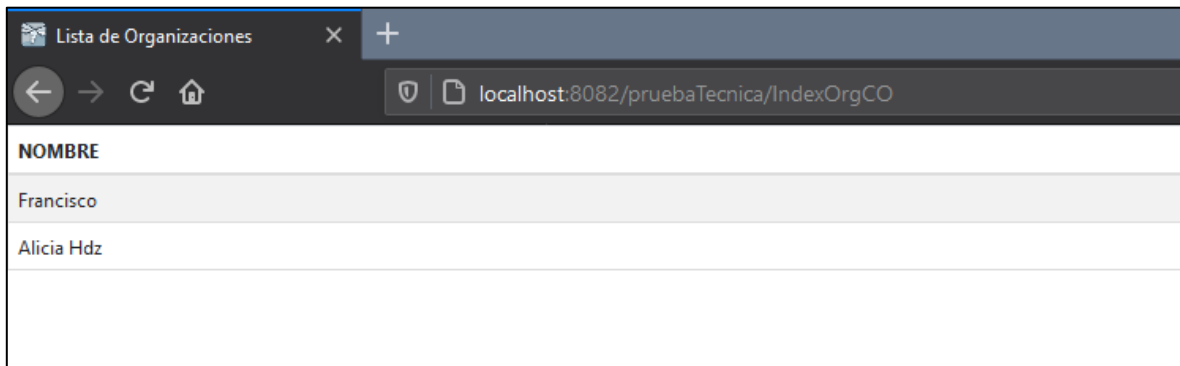
Revisando nuestro archivo **organizaciones.properties** podemos notar que se encuentra la nueva información.



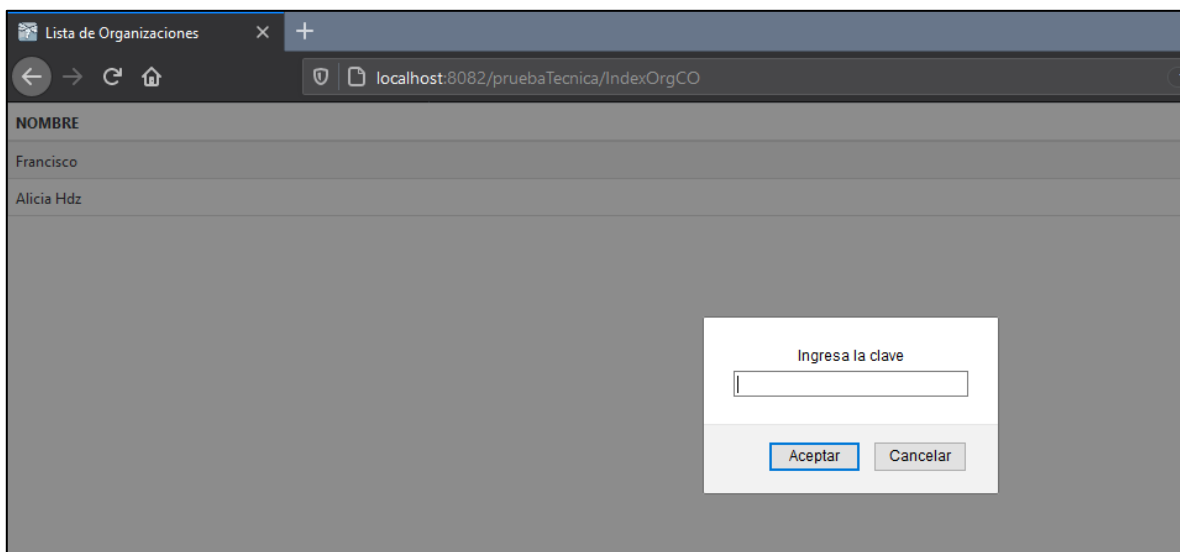
Para ver la lista de organizaciones que agregamos, damos clic en Lista de organizaciones la cual nos mandara a la vista con todos los nombres de las organizaciones



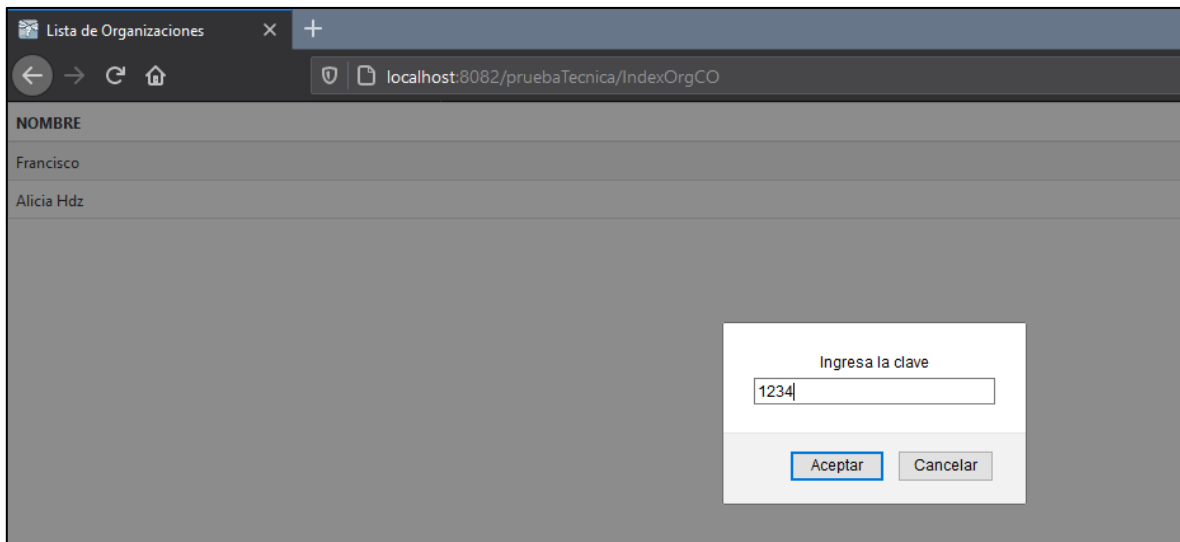
Nos muestra la lista de las organizaciones registradas.



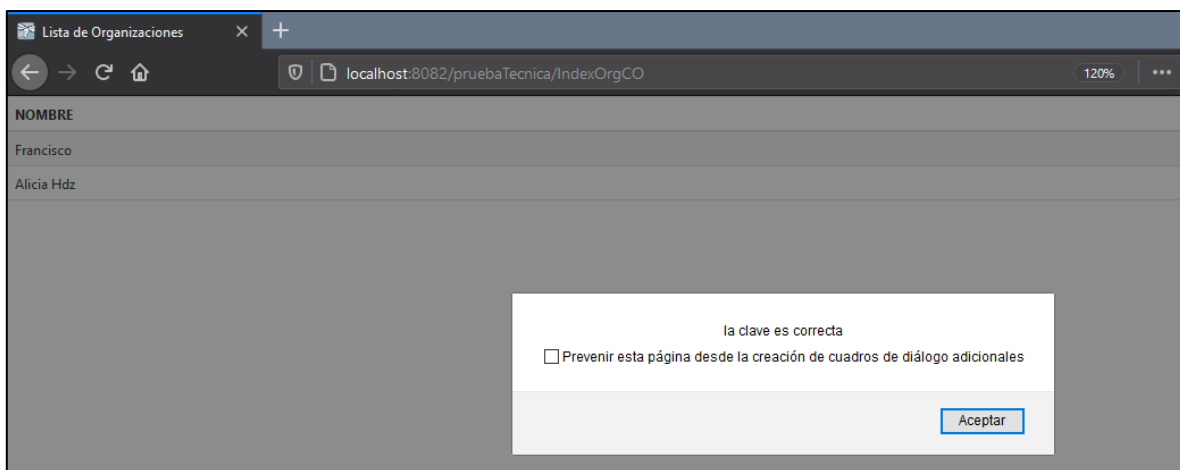
Al presionar sobre el nombre de la organización nos muestra el siguiente mensaje



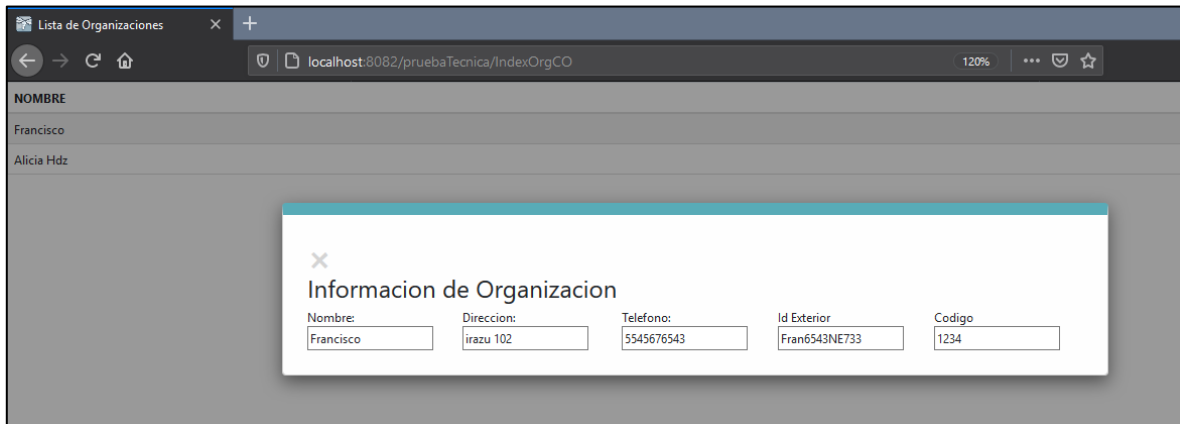
Ingresamos la clave de la organización en la que dimos clic.



Si la clave es correcta nos mostrara el siguiente mensaje:

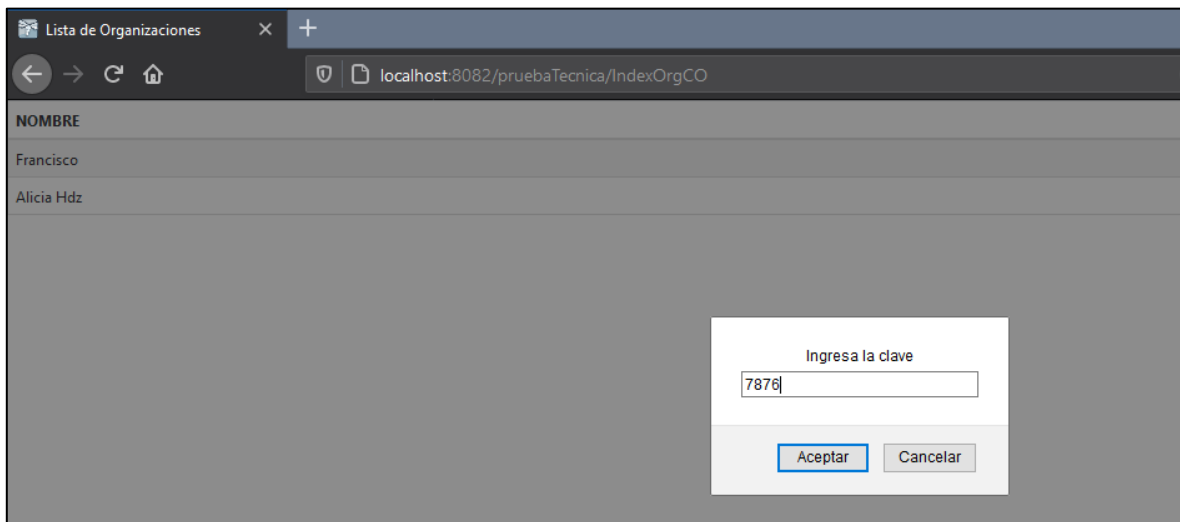


Al cerrar el mensaje anterior automáticamente nos abrirá la ventana modal con la información de la organización.

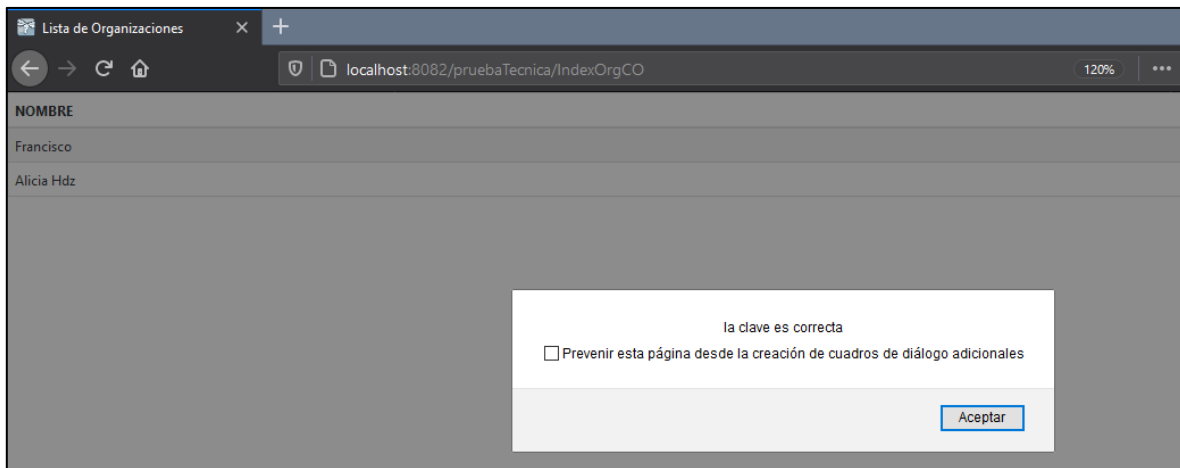


Realizamos el mismo procedimiento para la otra organización.

Ingresamos la clave:



Si la clave es correcta nos mostrara el siguiente mensaje:



Al cerrar la ventana anterior, automáticamente se abrirá la ventana modal con la información.

