



# UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

## MANUAL TÉCNICO

“IMPLEMENTACIÓN DE UN APLICATIVO RESPONSIVE PARA EL  
SEGUIMIENTO CLÍNICO DENTAL EN LA INSTITUCIÓN ODONTOLÓGICA  
CEMOC PROVINCIA DE PICHINCHA CANTÓN CAYAMBE.”

**AUTOR:** GUSTAVO RAUL CUZCO CUZCO

**DIRECTOR:** ING. DIEGO TREJO

**IBARRA – ECUADOR**

**2016**

# ÍNDICE DE CONTENIDO

ÍNDICE DE ILUSTRACIONES .....	III
<b>Anexo 1: Manual Técnico .....</b>	<b>4</b>
Objetivo .....	4
Herramientas Utilizadas.....	4
Código Fuente.....	4
JPA 6	
com.cemoc.ejb .....	13
com.cemoc.controller .....	15
@ManagedBean.....	16
@SessionScoped.....	16
WebPages (Boostrap 3.0) .....	24

# ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1: Raíz del Sistema SISCEMOC</i>	5
<i>Ilustración 2: Paquetes del sistema</i>	5
<i>Ilustración 3: Datos del WebPages</i>	6
<i>Ilustración 4: Creación de Objetos de la BDD con JPA</i>	7
<i>Ilustración 5: contenido del paquete facade</i>	13
<i>Ilustración 6: Beans del sistema</i>	16
<i>Ilustración 7: Vistas del sistema CEMOC</i>	24

## **Anexo 2: Manual Técnico**

En este anexo se explicara de manera más técnica el sistema de seguimiento clínico dental de la institución odontológica CEMOC.

### **Objetivo**

Puntualizar todas las entidades y códigos utilizados en cada una de ellas para el funcionamiento del Sistema informático de seguimiento clínico odontológico, tomando como lugar piloto a la Clínica Médica y Especialista Cayambe CEMOC en la provincia de Pichincha cantón Cayambe.

### **Herramientas Utilizadas**

- Herramienta de entorno de programación java
  - Net Beans IDE 8.0.1
- Contenedor de Aplicaciones Web
  - Apache Tomcat
- Base de Datos
  - PostgreSQL
- Vistas framewor
  - Bootstrap 3.0

### **Código Fuente**

Raíz del Proyecto, es donde se encuentra el resto del código del sistema.

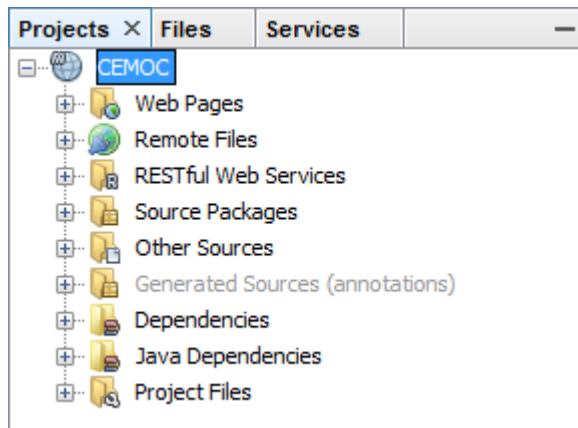


Ilustración 1: Raíz del Sistema SISCEMOC

Fuente: [Propia]

Dentro del mismo se componen de las siguientes partes:

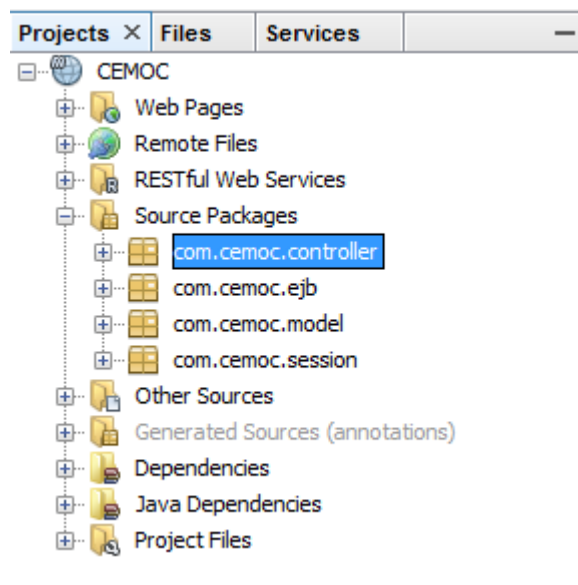
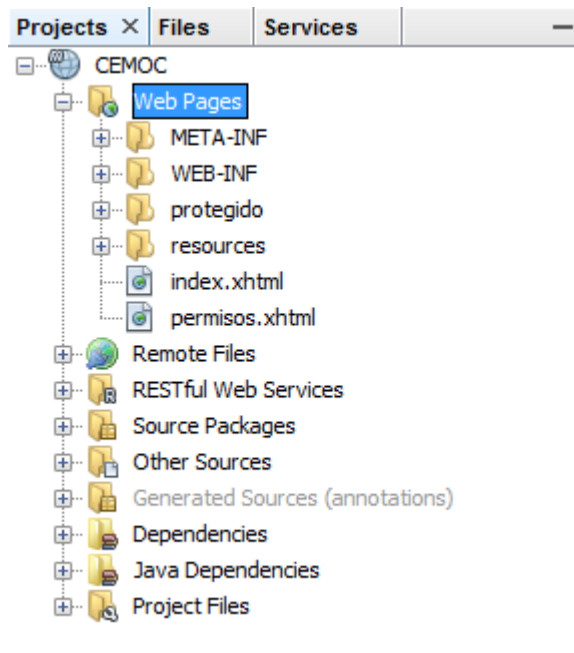


Ilustración 2: Paquetes del sistema

Fuente: Propia

En la carpeta WebPages se encuentran todas las plantillas de las vista, archivos de estilos, JavaScript, entre otros que permiten la presentación al usuario.



*Ilustración 3: Datos del WebPages*

Fuente: Propia

A continuación se detallara los contenidos que se tiene en cada uno de los paquetes de la carpeta JavaResources del proyecto.

## **JPA**

En el primer paquete tenemos las entidades que se lo hace mediante Java Persistence API, esta busca agrupar la manera en que funcionan los servicios que brinda un mapeo objeto relacional, con esto no se pierde la ventaja de orientada a objetos al interactuar con base de datos, además de permitir usar objetos POJO (clases simples java con métodos getter y setter básicos).

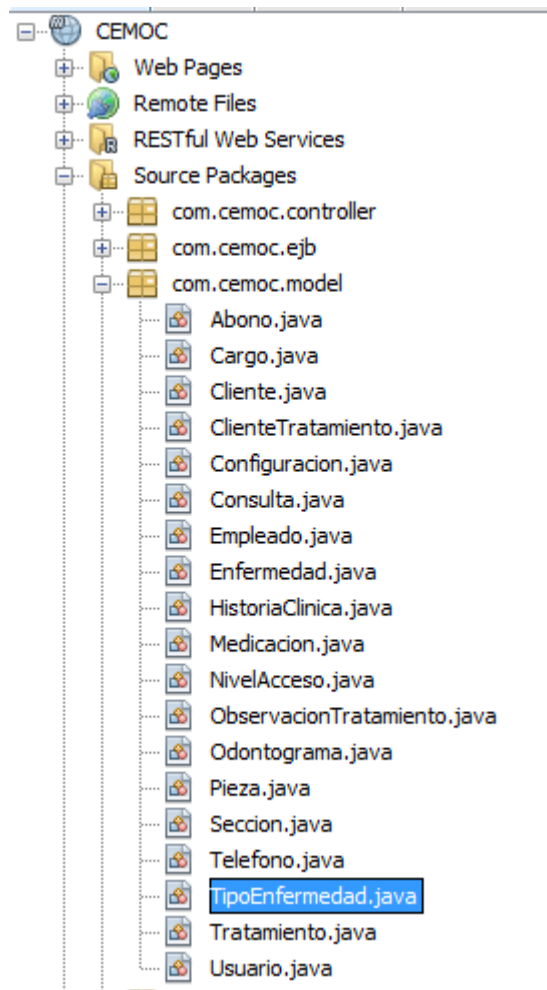


Ilustración 4: Creación de Objetos de la BDD con JPA

Fuente: Propia

Todas las entidades que se crea mediante el mapeo tienen la siguiente estructura así que se coloca un ejemplo a continuación.

```
package com.cemoc.model;
import java.io.Serializable;
import java.util.Date;
import java.util.List;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
```

```

import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;

@Entity
@Table(name = "cliente")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Cliente.findAll", query = "SELECT c FROM Cliente
c"),
    @NamedQuery(name = "Cliente.findByIdCliente", query = "SELECT c FROM
Cliente c WHERE c.idCliente = :idCliente"),
    @NamedQuery(name = "Cliente.findByIdentificacion", query = "SELECT c
FROM Cliente c WHERE c.identificacion = :identificacion"),
    @NamedQuery(name = "Cliente.findByIdFechaCreacion", query = "SELECT c
FROM Cliente c WHERE c.fechaCreacion = :fechaCreacion"),
    @NamedQuery(name = "Cliente.findByIdCorreo", query = "SELECT c FROM
Cliente c WHERE c.correo = :correo"),
    @NamedQuery(name = "Cliente.findByIdDireccion", query = "SELECT c FROM
Cliente c WHERE c.direccion = :direccion"),
    @NamedQuery(name = "Cliente.findByIdImagen", query = "SELECT c FROM
Cliente c WHERE c.imagen = :imagen"),
    @NamedQuery(name = "Cliente.findByIdEstado", query = "SELECT c FROM
Cliente c WHERE c.estado = :estado"),
    @NamedQuery(name = "Cliente.findByIdPrimerNombre", query = "SELECT c
FROM Cliente c WHERE c.primerNombre = :primerNombre"),
    @NamedQuery(name = "Cliente.findByIdSegundoNombre", query = "SELECT c
FROM Cliente c WHERE c.segundoNombre = :segundoNombre"),
    @NamedQuery(name = "Cliente.findByIdApellidoMaterno", query = "SELECT c
FROM Cliente c WHERE c.apellidoMaterno = :apellidoMaterno"),
    @NamedQuery(name = "Cliente.findByIdApellidoPaterno", query = "SELECT c
FROM Cliente c WHERE c.apellidoPaterno = :apellidoPaterno"))
public class Cliente implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id_cliente")
    private Integer idCliente;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 15)
    @Column(name = "identificacion")

```



```

private String identificacion;
@Basic(optional = false)
@NotNull
@Column(name = "fecha_creacion")
@Temporal(TemporalType.DATE)
private Date fechaCreacion;
@Size(max = 40)
@Column(name = "correo")
private String correo;
@Size(max = 120)
@Column(name = "direccion")
private String direccion;
@Size(max = 100)
@Column(name = "imagen")
private String imagen;
@Size(max = 1)
@Column(name = "estado")
private String estado;
@Size(max = 100)
@Column(name = "primer_nombre")
private String primerNombre;
@Size(max = 100)
@Column(name = "segundo_nombre")
private String segundoNombre;
@Size(max = 100)
@Column(name = "apellido_materno")
private String apellidoMaterno;
@Size(max = 100)
@Column(name = "apellido_paterno")
private String apellidoPaterno;
@OneToMany(mappedBy = "idCliente")
private List<ClienteTratamiento> clienteTratamientoList;
@OneToMany(mappedBy = "idCliente", cascade = CascadeType.ALL)
private List<HistoriaClinica> historiaClinicaList;
@OneToMany(mappedBy = "idCliente", cascade = CascadeType.ALL)
private List<Telefono> telefonoList;

public Cliente() {
}

public Cliente(Integer idCliente) {
    this.idCliente = idCliente;
}

public Cliente(Integer idCliente, String identificacion, Date fechaCreacion) {
    this.idCliente = idCliente;
    this.identificacion = identificacion;
    this.fechaCreacion = fechaCreacion;
}

```

```

public Integer getIdCliente() {
    return idCliente;
}

public void setIdCliente(Integer idCliente) {
    this.idCliente = idCliente;
}

public String getIdentificacion() {
    return identificacion;
}

public void setIdentificacion(String identificacion) {
    this.identificacion = identificacion;
}

public Date getFechaCreacion() {
    return fechaCreacion;
}

public void setFechaCreacion(Date fechaCreacion) {
    this.fechaCreacion = fechaCreacion;
}

public String getCorreo() {
    return correo;
}

public void setCorreo(String correo) {
    this.correo = correo;
}

public String getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public String getImagen() {
    return imagen;
}

public void setImagen(String imagen) {
    this.imagen = imagen;
}

public String getEstado() {
    return estado;
}

```

```

    }

    public void setEstado(String estado) {
        this.estado = estado;
    }

    public String getPrimerNombre() {
        return primerNombre;
    }

    public void setPrimerNombre(String primerNombre) {
        this.primerNombre = primerNombre;
    }

    public String getSegundoNombre() {
        return segundoNombre;
    }

    public void setSegundoNombre(String segundoNombre) {
        this.segundoNombre = segundoNombre;
    }

    public String getApellidoMaterno() {
        return apellidoMaterno;
    }

    public void setApellidoMaterno(String apellidoMaterno) {
        this.apellidoMaterno = apellidoMaterno;
    }

    public String getApellidoPaterno() {
        return apellidoPaterno;
    }

    public void setApellidoPaterno(String apellidoPaterno) {
        this.apellidoPaterno = apellidoPaterno;
    }

    @XmlTransient
    public List<ClienteTratamiento> getClienteTratamientoList() {
        return clienteTratamientoList;
    }

    public void setClienteTratamientoList(List<ClienteTratamiento>
clienteTratamientoList) {
        this.clienteTratamientoList = clienteTratamientoList;
    }

    @XmlTransient
    public List<HistoriaClinica> getHistoriaClinicaList() {

```

```

        return historiaClinicaList;
    }

    public void setHistoriaClinicaList(List<HistoriaClinica> historiaClinicaList) {
        this.historiaClinicaList = historiaClinicaList;
    }

    @XmlTransient
    public List<Telefono> getTelefonoList() {
        return telefonoList;
    }

    public void setTelefonoList(List<Telefono> telefonoList) {
        this.telefonoList = telefonoList;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idCliente != null ? idCliente.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Cliente)) {
            return false;
        }
        Cliente other = (Cliente) object;
        if ((this.idCliente == null && other.idCliente != null) || (this.idCliente != null
        && !this.idCliente.equals(other.idCliente))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.cemoc.model.Cliente[ idCliente=" + idCliente + " ]";
    }
}

```

## com.cemoc.ejb

En el siguiente paquete tiene todos los manejadores básicos los cuales permite interactuar con la base de datos. Los métodos más importantes que aquí se tienen son:

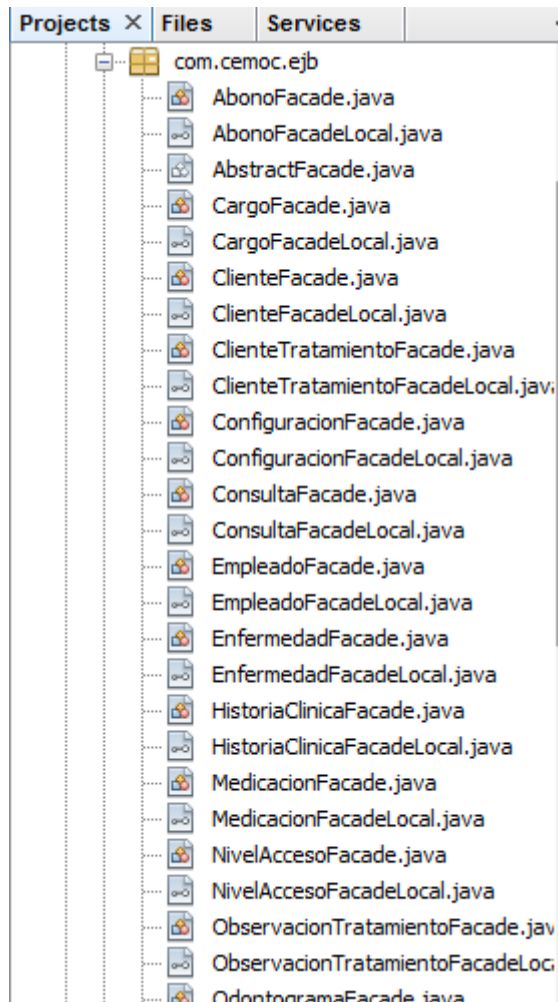


Ilustración 5: contenido del paquete facade

Fuente: Propia

- Insertar
- Eliminar
- Actualizar
- Buscar por ID

- Buscar por algún campo específico
- Lista filas de una tabla de la base de datos
- Listar datos de acuerdo a un ID

A continuación código de los manejadores básicos.

```
package com.cemoc.ejb;

import com.cemoc.model.Cliente;
import java.util.ArrayList;
import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

@Stateless
public class ClienteFacade extends AbstractFacade<Cliente> implements
ClienteFacadeLocal {
    @PersistenceContext(unitName = "cemocPU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public ClienteFacade() {
        super(Cliente.class);
    }

    /**
     * RETORNA UNA LISTA DE CLIENTES ACTIVOS
     * @return
     * @throws Exception
     */
    @Override
    public List<Cliente> listaClientesActivos() throws Exception {
        List<Cliente> lista = new ArrayList<>();
        try {
            String sql = "SELECT c "
                + " FROM Cliente c "
                + " WHERE c.estado = :estado "
                + " ORDER BY c.idCliente";
            Query query = em.createQuery(sql);
            query.setParameter("estado", "A");
        }
    }
}
```

```

        lista = query.getResultList();
    } catch (Exception e) {
        throw e;
    }
    return lista;
}
}
package com.cemoc.ejb;

import com.cemoc.model.Cliente;
import java.util.List;
import javax.ejb.Local;

@Local
public interface ClienteFacadeLocal {

    void create(Cliente cliente);

    void edit(Cliente cliente);

    void remove(Cliente cliente);

    Cliente find(Object id);

    List<Cliente> findAll();

    List<Cliente> findRange(int[] range);

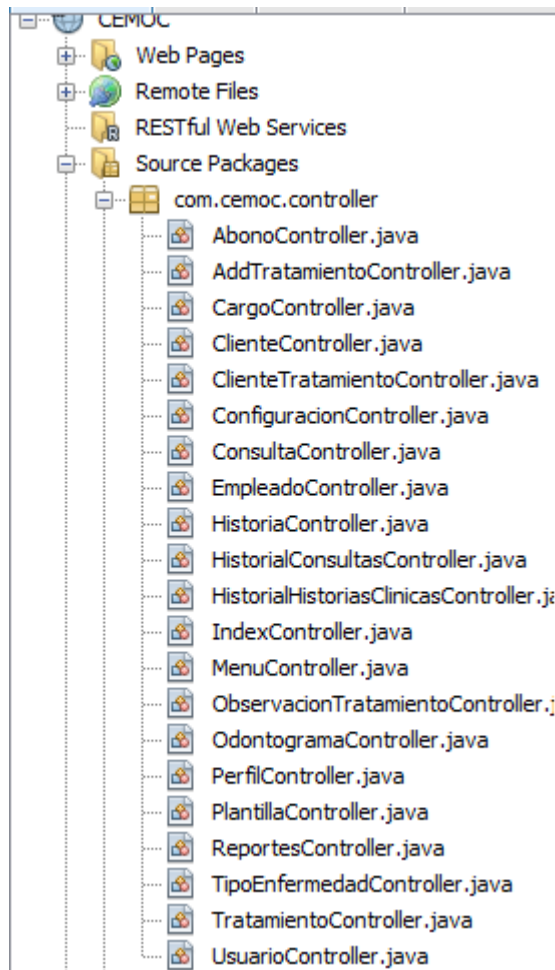
    int count();

    public List<Cliente> listaClientesActivos() throws Exception;
}

```

### **com.cemoc.controller**

Es el encargado del control de toda la presentación, aquí se consumen los métodos realizados anteriormente, es por ello que se dice que es el intermediario entre el Manejador de Objetos y el controlador, con esto podemos hacer al código mucho más reutilizable ya que de cualquier entidad controladora podemos llamar a los mismos métodos de crear departamento médico por ejemplo.



*Ilustración 6: Beans del sistema*

Fuente: Propia

### **@ManagedBean**

Ayuda a identificar y acceder desde el diseño, además permiten registrar las clases que van a ser Managed Beans. La anotación es `@javax.faces.bean.ManagedBean`, con esto ya no tenemos que especificar esta información en el archivo XML `faces-config.xml`, que tiene el proyecto creado.

### **@SessionScoped**

Nos permite manejar todo el proceso únicamente hasta cuando se termine la sesión actual.



```

package com.cemoc.controller;

import com.cemoc.ejb.AbonoFacadeLocal;
import com.cemoc.ejb.ClienteFacadeLocal;
import com.cemoc.ejb.ClienteTratamientoFacadeLocal;
import com.cemoc.model.Abono;
import com.cemoc.model.Cliente;
import com.cemoc.model.ClienteTratamiento;
import com.cemoc.model.Consulta;
import com.cemoc.session.SessionController;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import javax.annotation.PostConstruct;
import javax.ejb.EJB;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.view.ViewScoped;
import javax.inject.Inject;
import javax.inject.Named;

@ViewScoped
@Named
public class ClienteController implements Serializable {

    @EJB
    ClienteFacadeLocal clienteEJB;
    @EJB
    AbonoFacadeLocal abonoEJB;
    @EJB
    ClienteTratamientoFacadeLocal clienteTratEJB;

    @Inject
    Cliente cliente;
    @Inject
    ClienteTratamiento clienteTratamiento;
    @Inject
    Abono abono;

    List<Cliente> clientes;

    List<Cliente> clientesActivos;

    private List<Consulta> consultas;
    private List<ClienteTratamiento> clientesTratamientos;
    List<Abono> abonos;

    @Inject
    private SessionController session;

```

```

@PostConstruct
public void init() {
    getAllClientes();
    getClientesActivos();
    getAbonos();
}

public void getAllClientes() {
    clientes = clienteEJB.findAll();
}

public void getClientesActivos() {
    clientesActivos = new ArrayList<>();
    for (Cliente c : clientes) {
        if (c.getEstado().equals("A")) {
            clientesActivos.add(c);
        }
    }
}

public void getAllAbonos() {
    abonos = abonoEJB.findAll();
}

public void insertar() {
    try {
        if (cliente.getIdentificacion().isEmpty() || cliente.getIdentificacion() == null)
        {
            FacesContext.getCurrentInstance().addMessage(null,
                new FacesMessage(FacesMessage.SEVERITY_WARN, "Registro",
                "Ingrese una identificación!!!"));
            return;
        }
        if (cliente.getFechaCreacion() == null) {
            FacesContext.getCurrentInstance().addMessage(null,
                new FacesMessage(FacesMessage.SEVERITY_WARN, "Registro",
                "Ingrese una fecha!!!"));
            return;
        }
        if (cliente.getCorreo().isEmpty() || cliente.getCorreo() == null) {
            FacesContext.getCurrentInstance().addMessage(null,
                new FacesMessage(FacesMessage.SEVERITY_WARN, "Registro",
                "Ingrese un correo electronico!!!"));
            return;
        }
        if (cliente.getDireccion().isEmpty() || cliente.getDireccion() == null) {
            FacesContext.getCurrentInstance().addMessage(null,
                new FacesMessage(FacesMessage.SEVERITY_WARN, "Registro",
                "Ingrese una dirección!!!"));
            return;
        }
    }
}

```

```

    }

    if(!validadorDeCedula(cliente.getIdentificacion()))
    {
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Error en
la indentificación", "La Identificación es invalida."));
        return;
    }

    cliente.setEstado("A");
    clienteEJB.create(cliente);
    cliente = new Cliente();
    getAllClientes();
    FacesContext.getCurrentInstance().addMessage(null,
        new FacesMessage(FacesMessage.SEVERITY_INFO, "Registro",
"Correcto"));
    } catch (Exception e) {
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Registro",
"Error: " + e.getMessage()));
    }
}

public void leer(Cliente c) {
    cliente = c;
    if (tieneConsultas(c)) {
        consultas = c.getHistoriaClinicaList().get(0).getConsultaList();
    }
    if (tienePagos(c)) {
        clientesTratamientos = c.getClienteTratamientoList();
        abonos = c.getClienteTratamientoList().get(0).getAbonoList();
        System.out.println("id          cliente          "          +
clientesTratamientos.get(0).getIdCliente().getIdCliente());
    }
}

public void modificar() {
    try {
        clienteEJB.edit(cliente);
        getAllClientes();
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Edición",
"Correcto"));
    } catch (Exception e) {
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Edición",
"Error: " + e.getMessage()));
    } finally {

```

```

FacesContext.getCurrentInstance().getExternalContext().getFlash().setKeepMessages(true
);
    }
}

public void desactivar(Cliente c) {
    try {
        c.setEstado("I");
        clienteEJB.edit(c);
        getAllClientes();
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Edición",
"Correcto"));
    } catch (Exception ex) {
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Edición",
"Error: " + ex.getMessage()));
    } finally {

FacesContext.getCurrentInstance().getExternalContext().getFlash().setKeepMessages(true
);
    }
}

public void activar(Cliente c) {
    try {
        c.setEstado("A");
        clienteEJB.edit(c);
        getAllClientes();
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Edición",
"Correcto"));
    } catch (Exception ex) {
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Edición",
"Error: " + ex.getMessage()));
    } finally {

FacesContext.getCurrentInstance().getExternalContext().getFlash().setKeepMessages(true
);
    }
}

public String retornaEstado(Cliente c) {
    if (c.getEstado() != null && c.getEstado().equals("A")) {
        return "ACTIVO";
    }
}

```

```

        return "INACTIVO";
    }

    public boolean tieneHistoriaClinica(Cliente cliente) {
        if (cliente.getHistoriaClinicaList() != null &&
cliente.getHistoriaClinicaList().size() >= 1) {
            return true;
        }
        return false;
    }

    public boolean tieneConsultas(Cliente cliente) {
        if (cliente.getHistoriaClinicaList() != null &&
cliente.getHistoriaClinicaList().size() >= 1) {
            if (cliente.getHistoriaClinicaList().get(0).getConsultaList() != null) {
                return true;
            }
        }
        return false;
    }

    public boolean tieneTratamientos(Cliente cliente) {
        if (cliente.getClienteTratamientoList() != null &&
cliente.getClienteTratamientoList().size() >= 1) {
            return true;
        }
        return false;
    }

    public boolean tienePagos(Cliente cliente) {
        if (cliente.getClienteTratamientoList() != null &&
cliente.getClienteTratamientoList().size() >= 1) {
            if (cliente.getClienteTratamientoList().get(0).getAbonoList() != null) {
                return true;
            }
        }
        return false;
    }

    public boolean validadorDeCedula(String cedula) {
        boolean cedulaCorrecta = false;

        try {

            if (cedula.length() == 10) // ConstantesApp.LongitudCedula
            {
                int tercerDigito = Integer.parseInt(cedula.substring(2, 3));
                if (tercerDigito < 6) {
                    // Coeficientes de validación cédula
                    // El decimo digito se lo considera dígito verificador

```

```

        int[] coefValCedula = {2, 1, 2, 1, 2, 1, 2, 1, 2};
        int verificador = Integer.parseInt(cedula.substring(9, 10));
        int suma = 0;
        int digito = 0;
        for (int i = 0; i < (cedula.length() - 1); i++) {
            digito = Integer.parseInt(cedula.substring(i, i + 1)) *
coefValCedula[i];
            suma += ((digito % 10) + (digito / 10));
        }

        if ((suma % 10 == 0) && (suma % 10 == verificador)) {
            cedulaCorrecta = true;
        } else if ((10 - (suma % 10)) == verificador) {
            cedulaCorrecta = true;
        } else {
            cedulaCorrecta = false;
        }
    } else {
        cedulaCorrecta = false;
    }
} else {
    cedulaCorrecta = false;
}
} catch (NumberFormatException nfe) {
    cedulaCorrecta = false;
} catch (Exception err) {
    System.out.println("Una excepcion ocurrio en el proceso de validacion");
    cedulaCorrecta = false;
}

if (!cedulaCorrecta) {
    System.out.println("La Cédula ingresada es Incorrecta");
}
return cedulaCorrecta;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public List<Cliente> getClientes() {
    return clientes;
}

public void setClientes(List<Cliente> clientes) {
    this.clientes = clientes;
}

```

```

    }

    public List<Consulta> getConsultas() {
        return consultas;
    }

    public void setConsultas(List<Consulta> consultas) {
        this.consultas = consultas;
    }

    public List<ClienteTratamiento> getCientesTratamientos() {
        return clientesTratamientos;
    }

    public void setClientesTratamientos(List<ClienteTratamiento>
clientesTratamientos) {
        this.clientesTratamientos = clientesTratamientos;
    }

    public ClienteTratamiento getClienteTratamiento() {
        return clienteTratamiento;
    }

    public void setClienteTratamiento(ClienteTratamiento clienteTratamiento) {
        this.clienteTratamiento = clienteTratamiento;
    }

    public List<Abono> getAbonos() {
        return abonos;
    }

    public void setAbonos(List<Abono> abonos) {
        this.abonos = abonos;
    }

    public Abono getAbono() {
        return abono;
    }

    public void setAbono(Abono abono) {
        this.abono = abono;
    }
}

```

## WebPages (Bootstrap 3.0)

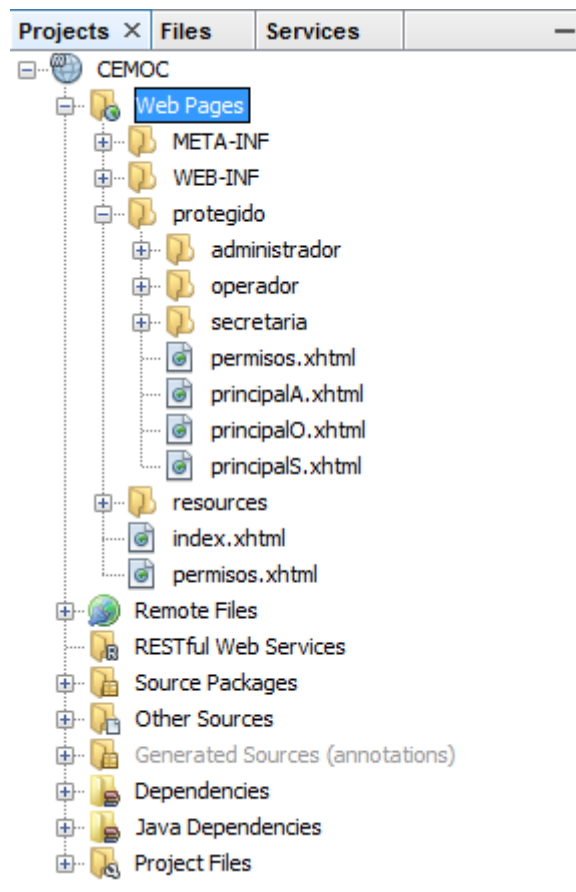


Ilustración 7: Vistas del sistema CEMOC

Fuente: Propia

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  xmlns:h="http://xmlns.jcp.org/jsf/html"
  xmlns:f="http://xmlns.jcp.org/jsf/core"
  xmlns:a="http://xmlns.jcp.org/jsf/passthrough">

<body>

  <ui:composition template="./principalA.xhtml">

    <ui:define name="content">
      <!-- Content Wrapper. Contains page content -->
      <div class="content-wrapper">
        <!-- Content Header (Page header) -->
```



```

<section class="content-header">
  <h1>
    Gestión de Clientes
    <small>Creación, listado y edicion de clientes.</small>
  </h1>
  <ol class="breadcrumb">
    <li><a href="/CEMOC/faces/protegido/principalA.xhtml"><i
class="fa fa-dashboard"></i> Inicio</a></li>
    <li class="active">Clientes</li>
  </ol>
</section>

<!-- Main content -->
<section class="content">
  <h:messages id="mensaje" showSummary="true" showDetail="true"
z-index: 1000;"
style="padding: 0; position: absolute; right: 20px; float: right;
errorClass="callout callout-danger"
warnClass="callout callout-warning" infoClass="callout
callout-info"/>

  <div class="row">
    <div class="box-group" id="accordion">
      <!-- we are adding the .panel class so bootstrap.js collapse plugin
detects it -->
      <div class="panel box box-primary">
        <div class="box-header with-border">
          <h4 class="box-title">
            <a data-toggle="collapse" data-parent="#accordion"
href="#collapseOne">
              Nuevo Cliente
            </a>
          </h4>
        </div>
        <div id="collapseOne" class="panel-collapse collapse">
          <div class="box-body">
            <h:form styleClass="validate-form">
              <h:messages id="mensaje" showSummary="true"
showDetail="true" style="padding: 0"
errorClass="callout callout-danger"
warnClass="callout callout-warning"
infoClass="callout callout-info"/>
            <div class="box-body">
              <div class="form-group">
                <label class="col-sm-2 control-
label">Cédula</label>
                <div class="col-sm-10">
                  <h:inputText
value="#{clienteController.cliente.identificacion}" styleClass="form-control ci"

```

```

                                a:placeholder="Documento de
identificación"/>
                                <span style="display:none" class="help-
block">El campo es inválido.</span>
                                </div>
                                <div class="clear"/>
                                </div>
                                <div class="form-group">
                                <label class="col-sm-2 control-label">Primer
Nombre</label>
                                <div class="col-sm-10">
                                <h:inputText
value="#{clienteController.cliente.primerNombre}" styleClass="form-control text"
                                a:placeholder="Primer nombre del
cliente"/>
                                <span style="display:none" class="help-
block">El campo es inválido. Solo texto</span>
                                </div>
                                <div class="clear"/>
                                </div>
                                <div class="form-group">
                                <label class="col-sm-2 control-label">Segundo
Nombre</label>
                                <div class="col-sm-10">
                                <h:inputText
value="#{clienteController.cliente.segundoNombre}" styleClass="form-control text"
                                a:placeholder="Segundo nombre del
cliente"/>
                                <span style="display:none" class="help-
block">El campo es inválido. Solo texto</span>
                                </div>
                                <div class="clear"/>
                                </div>
                                <div class="form-group">
                                <label class="col-sm-2 control-label">Apellido
Paterno</label>
                                <div class="col-sm-10">
                                <h:inputText
value="#{clienteController.cliente.apellidoPaterno}" styleClass="form-control text"
                                a:placeholder="Apellido paterno del
cliente"/>
                                <span style="display:none" class="help-
block">El campo es inválido. Solo texto</span>
                                </div>

```

```

        <div class="clear"/>
    </div>

    <div class="form-group">
        <label class="col-sm-2 control-label">Apellido
Materno</label>

        <div class="col-sm-10">
            <h:inputText
value="#{clienteController.cliente.apellidoMaterno}" styleClass="form-control text"
a:placeholder="Apellido materno del
cliente"/>
            <span style="display:none" class="help-
block">El campo es inválido. Solo texto</span>
        </div>
        <div class="clear"/>
    </div>
    <div class="form-group">
        <label class="col-sm-2 control-label">Fecha de
creación:</label>

        <div class="col-sm-10">
            <h:inputText
value="#{clienteController.cliente.fechaCreacion}"
styleClass="form-control datepicker-
input" a:placeholder="dd-MM-yyyy">
            <f:convertDateTime pattern="dd-MM-
yyyy" />
            </h:inputText>
        </div>
        <div class="clear"/>
    </div>
    <div class="form-group">
        <label class="col-sm-2 control-
label">Email</label>

        <div class="col-sm-10">
            <h:inputText
value="#{clienteController.cliente.correo}" styleClass="form-control mail"
a:placeholder="Correo electrónico"/>
            <span style="display:none" class="help-
block">El campo es inválido.</span>
        </div>
        <div class="clear"/>
    </div>
    <div class="form-group">
        <label class="col-sm-2 control-
label">Dirección</label>

        <div class="col-sm-10">

```

```

        <h:inputTextarea
value="#{clienteController.cliente.direccion}" styleClass="form-control allText"
        a:placeholder="Dirección
domiciliaria"/>
        <span style="display:none" class="help-
block">El campo es inválido. Solo texto y números</span>
        </div>
        <div class="clear"/>
        </div>
        <div class="box-footer">
        <h:commandLink styleClass="btn btn-success pull-
right submit"
        action="#{clienteController.insertar()}">
        <span class="glyphicon glyphicon-floppy-
disk"></span> Guardar
        </h:commandLink>
        </div>
        </h:form>
        </div>
        </div>
        </div>
        <div class="panel box box-success">
        <div class="box-header with-border">
        <h4 class="box-title">
        <a data-toggle="collapse" data-parent="#accordion"
href="#collapseThree">
        Lista de Clientes
        </a>
        </h4>
        </div>
        <div id="collapseThree" class="panel-collapse collapse in">
        <div class="box-body">
        <h:form id="tblClientes">
        <table id="example1" class="table table-bordered
table-striped">
        <thead>
        <tr>
        <th>Cédula</th>
        <th>Nombres</th>
        <th>Apellidos</th>
        <th>Fecha de creación</th>
        <th>Correo electrónico</th>
        <th>Dirección</th>
        <th>Estado</th>
        <th>Acciones</th>
        </tr>
        </thead>
        <tbody>

```

```

                                <ui:repeat                                var="o"
value="#{clienteController.clientes}" varStatus="status">
                                <tr>
                                <td>#{o.identificacion}</td>
                                <td>#{o.primerNombre}
                                <td>#{o.apellidoPaterno}
                                <td>
                                <h:outputText
                                <f:convertDateTime pattern="dd-MM-
                                yyyy" />
                                </h:outputText>
                                </td>
                                <td>#{o.correo}</td>
                                <td>#{o.direccion}</td>
                                <td>#{clienteController.retornaEstado(o)}</td>
                                <td>
                                <div class="btn-group">
                                <h:commandLink styleClass="btn btn-
                                primary pull-left"
                                action="#{clienteController.leer(o)}"
                                a:data-target="#myModal">
                                <f:ajax                                render=":frmEditar
                                :mensaje"/>
                                <span class="glyphicon glyphicon-
                                pencil"></span> Modificar
                                </h:commandLink>
                                <h:commandLink styleClass="btn btn-
                                warning pull-right"
                                action="#{clienteController.desactivar(o)}"
                                rendered="#{o.estado eq
                                'A'}">
                                <f:ajax                                render="tblClientes
                                :mensaje"/>
                                <span class="glyphicon glyphicon-
                                minus"></span> Desactivar
                                </h:commandLink>
                                <h:commandLink styleClass="btn btn-
                                success pull-right"
                                action="#{clienteController.activar(o)}"
                                rendered="#{o.estado eq 'I'}"
                                >

```

```

:mensaje"/>
check"></span> Activar
</div>
</h:commandLink>
</div>
</td>
</tr>
</ui:repeat>
</tbody>
</table>
</h:form>
<!--modal-->
<div class="modal fade" id="myModal" role="dialog">
  <div class="modal-dialog modal-sm">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-
dismiss="modal">&times;</button>
        <h4 class="modal-title">Edición de
clientes</h4>
      </div>
      <h:form id="frmEditar">
        <div class="modal-body">
          <div class="box-body">
            <div class="form-group">
              <label class="col-sm-2 control-
label">Cédula</label>
              <div class="col-sm-10">
                <h:inputText
value="#{clienteController.cliente.identificacion}" styleClass="form-control"
a:placeholder="Documento
de identificación"/>
              </div>
            </div>
            <div class="clear"/>
          </div>
          <div class="form-group">
            <label class="col-sm-2 control-
label">Fecha de creación:</label>
            <div class="col-sm-10">
              <h:inputText
value="#{clienteController.cliente.fechaCreacion}"
styleClass="form-control
datepicker-input" a:placeholder="dd-MM-yyyy">
              <f:convertDateTime pattern="dd-
MM-yyyy" />
            </div>
          </div>
        </h:inputText>

```

```

        </div>
        <div class="clear"/>
    </div>
    <div class="form-group">
        <label class="col-sm-2 control-
label">Email</label>

        <div class="col-sm-10">
            <h:inputText
value="#{clienteController.cliente.correo}" styleClass="form-control"
a:placeholder="Correo
electrónico"/>

            </div>
            <div class="clear"/>
        </div>
        <div class="form-group">
            <label class="col-sm-2 control-
label">Dirección</label>

            <div class="col-sm-10">
                <h:inputTextarea
value="#{clienteController.cliente.direccion}" styleClass="form-control"
a:placeholder="Dirección
domiciliaria"/>

                </div>
                <div class="clear"/>
            </div>
        </div>

    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-
default" data-dismiss="modal">
remove"></span>Close</button>
        <h:commandLink type="button" class="btn
btn-primary" a:data-dismiss="modal"
action="#{clienteController.modificar()}">
execute="@form"/>
        <f:ajax render=":tblClientes :mensaje"
disk"></span>Guardar</h:commandLink>
    </div>
</h:form>
</div>
</div>
</div>
<!--fin modal-->

```

```
        </div>
      </div>
    </div>
  </div>
</div>
</section><!-- /.content -->
</div><!-- /.content-wrapper -->

</ui:define>

</ui:composition>

</body>
</html>
```