

CARRERA: Ingeniería de Sistemas

ASIGNATURA: Aplicaciones Distribuidas

Nombres:

Luis Ramon
Xavier Llauca
Mauricio Mocha

Tema:

1

TÍTULO: Revisión de anteproyecto “eventos , fiestas populares y tradicionales”

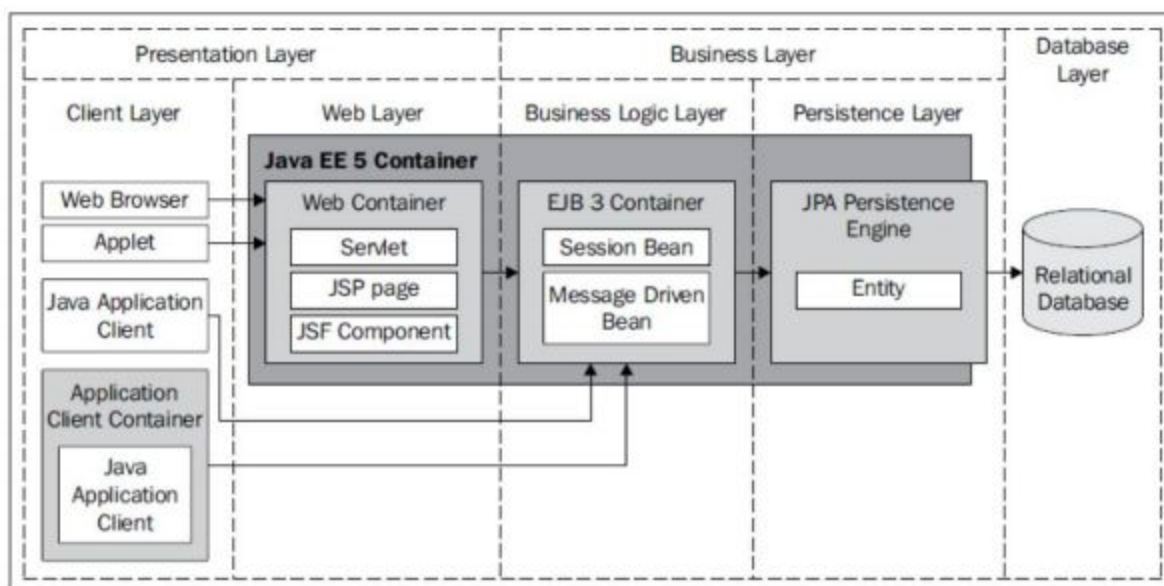
OBJETIVO.

- funcionalidades del sistema de administración

ACTIVIDADES POR DESARROLLAR

- Arquitectura JEE
- Calidad de código (convenciones de código java y documentación).
- Validaciones y control de excepciones.
- Presentación visual de la aplicación web.
- Bitácora detallada del trabajo realizado (documento en pdf).
- Diagrama de clases completo.

ARQUITECTUR JEE



Paquetes distribuidos en la arquitectura jee

Paquete ups.edu.ec.modelo

Paquete ups.edu.ec.controlador

Paquete ups.edu.ec.datos

Capa Cliente

Esta capa corresponde a lo que se encuentra en el computador del cliente. Es la interfaz grafica del sistema y se encarga de interactuar con el usuario. J2EE tiene soporte para diferentes tipos de clientes incluyendo clientes HTML, applets Java y aplicaciones Java

Capa Web

Se encuentra en el servidor web y contiene la lógica de presentación que se utiliza para generar una respuesta al cliente. Recibe los datos del usuario desde la capa cliente y basado en estos genera una respuesta apropiada a la solicitud.

Capa Negocio

Se encuentra en el servidor de aplicaciones y contiene el núcleo de la lógica del negocio de la aplicación. Provee las interfaces necesarias para utilizar el servicio de componentes del negocio. Las componentes del negocio interactúan con la capa de datos y son típicamente implementadas como componentes EJB.

Componentes EJB

La arquitectura EJB es una tecnología de lado del servidor para desarrollo y ejecución del servidor para desarrollo y ejecución de componentes para lógica de negocio de aplicación empresarial. También hay que destacar que los componentes EJB son escalables, transaccionales y multiusuario.

Clasificación de componentes EJB

1.- Session Bean:

provee un servicio detrás de un cliente y usualmente durante una única sesión. Ejecuta operaciones de cálculos o accesos a DB para el cliente.

2.- Entit y Bean

: un bean de entidad es un objeto persistente que representa el mantenimiento de los datos en un almacenamiento.

Capa Datos

Esta capa es responsable del sistema de información de la empresa o Enterprise InformationSystem (EIS) que incluye bases de datos, sistema de procesamiento datos, sistemas legados y sistemas de planificación de recursos. Esta capa es el punto donde las aplicaciones J2EE se integran con otros sistemas no J2EE o con sistemas legados.

CALIDAD DE CÓDIGO

Documentación de código clase Artista perteneciente al paquete modelo

```

1 package ups.edu.ec.modelo;
2 //IMPORT
3 import javax.persistence.Column;
4 import javax.persistence.Entity;
5 import javax.persistence.Id;
6 import javax.validation.constraints.NotNull;
7 import javax.validation.constraints.Size;
8
9 @Entity
10 public class Artista {
11
12     // DECLARACION DE ATRIBUSTOS DE ARTISTA
13     @Id
14     private int codigo;
15
16     @NotNull
17     @Column(name="art_nombre", length=10)
18     @Size(min=6,max=25)
19     private String nombre;
20
21     //GENERACION DE GET & SET DE CADA ATRIBUTO DE LA TABLA
22
23     public int getCodigo() {
24         return codigo;
25     }
26
27     public void setCodigo(int codigo) {
28         this.codigo = codigo;
29     }
30
31     public String getNombre() {
32         return nombre;
33     }
34
35     public void setNombre(String nombre) {
36         this.nombre = nombre;
37     }
38
39     //GENERACION DE TOSTRING PARA CVISUALIZACION Y COMPROBACION DE PASO DE PARAMETROS DE LA ENT
40
41     @Override
42     public String toString() {
43         return "Artista [codigo=" + codigo + ", nombre=" + nombre + "]";
44     }
45
46
47

```

Documentación de código clase Artista perteneciente al paquete modeloDAO

```

1 package ups.edu.ec.datos;
2
3 import java.util.List;
4
5 import javax.ejb.Stateless;
6 import javax.inject.Inject;
7 import javax.persistence.EntityManager;
8 import javax.persistence.Query;
9
10 import ups.edu.ec.modelo.Artista;
11
12 @Stateless
13 public class ArtistaDAO {
14
15     @Inject
16     private EntityManager em;
17
18     //CREACION DE METODO GUARDAR
19     public void guardar(Artista art) {
20         //LECTURA DE CODIGO DE ARTISTA
21         Artista aux = leer(art.getCodigo());
22         //COMPARACION SI VARIABLE CREADA DIFERENTE O IGUAL AL CODIGO INGRESADO
23         //ENTONCES ACTUALIZA CASO CONTRARIO INSERTA
24         if(aux!=null){
25
26             actualizar(art);
27
28         }else {
29             insertar(art);
30         }
31     }
32
33     //CREACION DE METODO INSERTAR
34     public void insertar (Artista art) {
35
36         em.persist(art);
37     }
38
39     //CREACION DE METODO ACTUALIZAR QUE NOS PERMITIRA ACTULIZAR NUESTRO ARTISTA CREADO
40     public void actualizar(Artista art) {
41
42         em.merge(art);
43     }
44
45     //CREACION DE METODO LEER QUE NOS PERMITIRA LEER NUESTRO ARTISTA CREADO
46     public Artista leer(int id) {
47         Artista art = em.find(Artista.class, id);

```

Paquete ups.edu.ec.modelo

- clases que contiene el paquete
 - Artista
 - Asistente
 - Canton
 - Categoria

- Concurso
- Evento
- Gastronomía
- Género
- Parroquia
- Sector

Paquete ups.edu.ec.datos

- **clases que contiene el paquete**
 - ArtistaDAO
 - AsistenteDAO
 - CantonDAO
 - CategoriaDAO
 - ConcursoDAO
 - EventoDAO
 - GastronomíaDAO
 - GéneroDAO
 - ParroquiaDAO
 - SectorDAO

Paquete ups.edu.ec.controlador

- **clases que contiene el paquete**
 - ArtistaControlador
 - AsistenteControlador
 - CantonControlador
 - CategoriaControlador
 - ConcursoControlador
 - EventoControlador
 - GastronomíaControlador
 - GéneroControlador
 - ParroquiaControlador
 - SectorControlador

Detalle de clases pertenecientes al paquete ups.edu.ec.modelo

- **Artista**

| codigo | Nombre |
|--------|---------------|
| 1 | Gerardo Moran |

| | |
|---|---------------|
| 2 | Adrian Lucia |
| 3 | Pamela Cortes |

- **Asistente**

| codigo | nombre | cedula | teléfono | direccio | usuario | password |
|--------|------------------|--------------|------------|-------------|------------|-------------|
| 1 | Carlos Samaniego | 010534872251 | 0985312356 | Ricaurte | csamaniego | samaniego21 |
| 2 | Andres Velez | 0163211457 | 0952345211 | Totoracocha | avelez | velez21 |
| 3 | Jose Arteaga | 0177823457 | 0963561235 | el cebollar | jarteaga | arteaga21 |

- **Canton**

| codigo | nombre |
|--------|----------|
| 1 | Cuenca |
| 2 | Gualaceo |
| 3 | Nabon |

- **Categoria**

| codigo | nombre | descripción |
|--------|-----------------|---|
| 1 | fiestas popular | Fiesta popular por la parroquializacion de Luis Cordero (Laguán) |
| 2 | Evento social | Matrimonio de la familia Sanchez-Cardenas |
| 3 | Expo feria | Expo feria realizada por el sindicato de choferes profesionales de Cuenca |

- **Concurso**

| codigo | nombre | descripción |
|---------------|---|---|
| 1 | concurso de danza | Concurso de danzas y premiación a los mejores grupos |
| 2 | la mejor pareja | Premio a la mejor pareja y a los que mejor bailan |
| 3 | reconocimiento de las marcas de vehiculos | premio a las personas que puedan reconocer las diferentes marcas de vehiculos |

- **Evento**

| codigo | fecha | descripción | representante |
|---------------|--------------|--|----------------------|
| 1 | 2017/12/15 | Fiesta popular por la celebración del carnaval | Ezequiel Ramirez |
| 2 | 2018/01/19 | Fiestas de Cuenca | Carina Alvear |
| 3 | 2018/03/02 | Concierto por aniversario de radio 103.30 fm | Sandra Salir |

- **Gastronomía**

| codigo | descripción |
|---------------|---------------------------------------|
| 1 | venta libre de productos alimenticios |
| 2 | venta de platos típico de la ciudad |
| 3 | venta de los mejores helados |

- **Género**

| codigo | nombre |
|---------------|---------------|
|---------------|---------------|

| | |
|----------|-------------------|
| 1 | Pasillos |
| 2 | Vallenatos |
| 3 | Baladas |

- **Parroquia**

| codigo | nombre | codigo_postal |
|---------------|------------------------------|----------------------|
| 1 | Checa | 014439 |
| 2 | Luis Cordero (Laguan) | 124343 |
| 3 | El progreso | 554327 |

- **Sector**

| codigo | nombre | direccion |
|---------------|--|---|
| 1 | centro de la parroquia checa | acceso principal a la parroquia Checa |
| 2 | centro de la parroquia Luis Cordero | acceso principal a la parroquia Luis Cordero |
| 3 | centro de la parroquia el Progreso | acceso principal a la parroquia el Progreso |

Detalle de clases pertenecientes al paquete ups.edu.ec.datos

- **ArtistaDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|-----------------|------------------|--------------------|-----------------|-----------------|-----------------|
| guardar Artista | insertar Artista | actualizar Artista | leer Artista | borrar Artista | Listado Artista |

- **AsistenteDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|-------------------|--------------------|----------------------|----------------|------------------|-------------------|
| guardar Asistente | insertar Asistente | actualizar Asistente | leer Asistente | borrar Asistente | Listado Asistente |

- **CantonDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|----------------|-----------------|-------------------|-------------|---------------|----------------|
| guardar Canton | insertar Canton | actualizar Canton | leer Canton | borrar Canton | Listado Canton |

- **CategoriaDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|-------------------|--------------------|----------------------|----------------|------------------|-------------------|
| guardar Categoria | insertar Categoria | actualizar Categoria | leer Categoria | borrar Categoria | Listado Categoria |

- **ConcursoDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|------------------|-------------------|---------------------|---------------|-----------------|------------------|
| guardar Concurso | insertar Concurso | actualizar Concurso | leer Concurso | borrar Concurso | Listado Concurso |

- **EventoDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|----------------|-----------------|-------------------|-------------|---------------|----------------|
| guardar Evento | insertar Evento | actualizar Evento | leer Evento | borrar Evento | Listado Evento |

- **GastronomíaDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|---------------------|----------------------|------------------------|------------------|--------------------|---------------------|
| guardar Gastronomía | insertar Gastronomía | actualizar Gastronomía | leer Gastronomía | borrar Gastronomía | Listado Gastronomía |

- **GéneroDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|----------------|-----------------|-------------------|-------------|---------------|----------------|
| guardar Género | insertar Género | actualizar Género | leer Género | borrar Género | Listado Género |

- **ParroquiaDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|-------------------|--------------------|----------------------|----------------|------------------|-------------------|
| guardar Parroquia | insertar Parroquia | actualizar Parroquia | leer Parroquia | borrar Parroquia | Listado Parroquia |

- **SectorDAO**

| metodo 1 | metodo 2 | metodo 3 | metodo 4 | metodo 5 | metodo 6 |
|----------------|-----------------|-------------------|-------------|---------------|----------------|
| guardar Sector | insertar Sector | actualizar Sector | leer Sector | borrar Sector | Listado Sector |

Detalle de clases pertenecientes al paquete ups.edu.ec.controlador

- **ArtistaControlador**

| Action controller |
|-------------------|
| guardad Artista |

- **AsistenteControlador**
-

| Action controller |
|-------------------|
| guardad Asistente |

- **CantonControlador**
-

| Action controller |
|-------------------|
| guardad Canton |

- **CategoriaControlador**

| Action controller |
|-------------------|
| guardad Categoria |

- **ConcursoControlador**
-

| |
|--------------------------|
| Action controller |
|--------------------------|

| |
|------------------|
| guardad Concurso |
|------------------|

- **EventoControlador**
-

| |
|--------------------------|
| Action controller |
|--------------------------|

| |
|----------------|
| guardad Evento |
|----------------|

- **GastronomíaControlador**

| |
|--------------------------|
| Action controller |
|--------------------------|

| |
|---------------------|
| guardad Gastronomía |
|---------------------|

- **GéneroControlador**

| |
|--------------------------|
| Action controller |
|--------------------------|

| |
|----------------|
| guardad Genero |
|----------------|

- **ParroquiaControlador**

| |
|--------------------------|
| Action controller |
|--------------------------|

| |
|------------------|
| guardadParroquia |
|------------------|

- **SectorControlador**

| |
|--------------------------|
| Action controller |
|--------------------------|

| |
|----------------|
| guardad Sector |
|----------------|

Fallos

@NotNull

Esta anotación indicará que el valor de la propiedad a la que se le aplica no puede ser null

@Size

Establece los límites inclusivos de `@Max` y `@Min` a la longitud de la `CharSequence` (`String`), el número de valores de la `Collection`, el número de entradas en un `Map`, o el número de elementos de un array de cualquier tipo.

@Column

Se aplica a una propiedad Java e indica el nombre de la columna de la base de datos en la que se persistirá la propiedad.

5- descripción de lo ocurrido

VALIDACIONES Y CONTROL DE EXCEPCIONES

Las restricciones de validación son anotaciones o código XML que se añaden a una clase, un campo o método de un componente de JavaBeans. Las restricciones las puede crear o definir el usuario. Se utilizan para definir las definiciones de restricciones regulares y para crear las restricciones. Las restricciones incorporadas las define la especificación de validación de beans y están disponibles con cada proveedor de validación.

import javax.validation.constraints.NotNull;

`@NotNull` asegura que se especifica un elemento

Todas las restricciones tienen una lógica de validación correspondiente que se inicia en tiempo de ejecución cuando se valida la entidad. Si no se cumple alguna de las restricciones, el JPA lanza una excepción `ConstraintViolationException`

import javax.validation.constraints.Size;

Comprueba si el tamaño del elemento anotado está entre min y max (inclusive). La longitud de la columna se establecerá en máx.

Docente / Técnico Docente: Ing. Cristian Timbi

Firma Docente: _____

Nombre de estudiante: _____

Firma de estudiante: _____

Nombre de estudiante: _____

Firma de estudiante: _____

Nombre de estudiante: _____

Firma de estudiante: _____