

Sistema de Control de Energía Eléctrica

Manual Técnico

09/11/2015

YACHAY.EP

Autora: Jenny Rea



ÍNDICE

DESCRIPCIÓN DEL SISTEMA	4
HERRAMIENTAS APLICADAS	4
NAVEGADORES	5
FRAMEWORKS.....	5
DIAGRAMA ENTIDAD-RELACIÓN.....	6
DESCRIPCIÓN DE LAS CLASES Y MÉTODOS	7
Package controlEnergiaEP.model.dao.entities.....	7
Class Parametro	7
Class Tipousuario	8
Class Funcionario	9
Class Evento	10
Class Modelo	11
Class Raspberry	12
Class Tipodispositivo	13
Class Dispositivo	14
Class Actividad	15
Class Registro	16
Aplicación Web	17
Package controlEnergiaEP.model.manager.....	17
Class ManagerDAO	17
Class ManagerActividad	20
Class ManagerCalendario	21
Class ManagerControlEnergia	22
Class ManagerReportes	29
Package controlEnergiaEP.controller.....	30
Class BeanAplicacion	30
Class BeanCalendario	31
Class BeanDispositivo	32
Class BeanEventos	33

Class BeanFuncionario	34
Class BeanLogin	35
Class BeanModelo	36
Class BeanParametros	36
Class BeanRaspberry	37
Class BeanReporte	37
Class BeanTipoDispositivo	39
Class ConexionRaspberry	40
Class MailUtil	40
Class JSFUtil	41
Aplicación de Escritorio TIMER	42
Package controlEnergiaEP.model.manager	42
Class ManagerControlEnergia	42
Package controlEnergiaEP.model.controller	45
Class BeanTimer	45
Package vista	46
Class VistaTimer	46
Tabla parámetro	48
Tabla tipousuario	48
Tabla funcionario	49
Tabla evento	50
Tabla modelo	50
Tabla raspberry	51
Tabla tipodispositivo	52
Tabla dispositivo	52
Tabla actividad	53
Tabla registro	54

DESCRIPCIÓN DEL SISTEMA

Tras la necesidad de automatizar el paso u obstrucción del flujo de corriente eléctrica para encender o apagar los dispositivos (depende del equipo) en las oficinas administrativas de Yachay EP. (Departamento de Soporte y Operaciones Tecnológicas); se crea el presente sistema: "SISTEMA DE CONTROL DE ENERGÍA ELÉCTRICA". A través del presente se implementa el CRUD de: funcionarios, eventos, parámetros, raspberries, modelos raspberry, dispositivos, tipo de dispositivos y actividades. Adicional a lo mencionado anteriormente permite la generación de reportes del consumo de dispositivos, evaluado en un rango de fechas seleccionadas previamente. Referente a la ejecución de las actividades creadas por el sistema se crea el programa de escritorio "TIMER" el cual envía señales al equipo Raspberry PI.

HERRAMIENTAS APLICADAS

JDK 7.0

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

POSTGRES 9.0

Conjuntamente con Eclipse se hace uso del SGBD, que en este caso es Postgres. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

APACHE TOMCAT 7.055

El servidor utilizado es Apache Tomcat que es una implementación de código abierto de software de las tecnologías Java Servlet y JavaServer Pages.

ECLIPSE

Para desarrollar la aplicación se utiliza Eclipse IDE JEE Kepler R2. Herramienta de programación de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido".

NAVEGADORES

Mozilla Firefox

Navegador web libre y de código abierto desarrollado para Microsoft Windows, Mac OS X y GNU/Linux. Usa el motor Gecko para renderizar páginas webs, el cual implementa actuales y futuros estándares web.

Google Chrome

Navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto, como el motor de renderizado Blink (bifurcación o fork de WebKit).

Internet Explorer 10

Navegador que está disponible para Windows 7 SP1, Windows 8 y Windows 8.1. Esta nueva versión incorpora considerables avances en la interpretación de estándares web respecto a sus precursores, como el soporte para CSS3, SVG, HTML5.

FRAMEWORKS

PrimeFaces

Librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces.

JPA Entities

Framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).

DIAGRAMA ENTIDAD-RELACIÓN

El diseño entidad – relación de la base de datos está diseñada en BrModel. En la Figura 1., se visualiza el mismo.

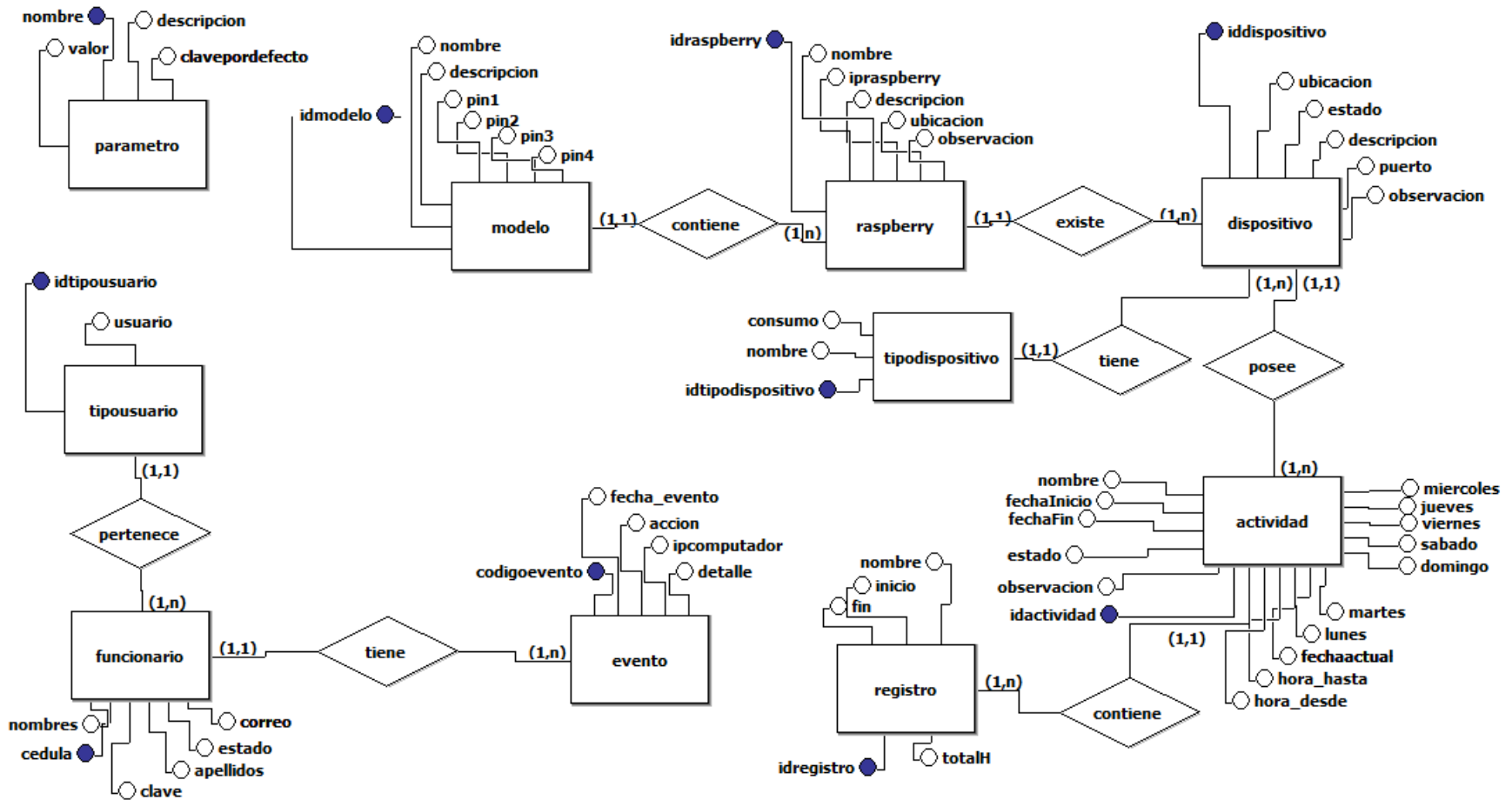


Figura 1.

DESCRIPCIÓN DE LAS CLASES Y MÉTODOS

Package controlEnergiaEP.model.dao.entities

Class Parametro

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Parametro

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Parametro**

extends java.lang.Object

implements java.io.Serializable

Serialized Form

Constructor Summary

Constructors
Constructor and Description
<u>Parametro()</u>

Method Summary

Methods	
Modifier and Type	Method and Description
java.lang.String	<u>getClavepordefecto()</u>
java.lang.String	<u>getDescripcion()</u>
java.lang.String	<u>getNombre()</u>
java.lang.String	<u>getValor()</u>
void	<u>setClavepordefecto()</u> (java.lang.String clavepordefecto)
void	<u>setDescripcion()</u> (java.lang.String descripcion)
void	<u>setNombre()</u> (java.lang.String nombre)
void	<u>setValor()</u> (java.lang.String valor)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Tipousuario

java.lang.Object

└─ controlEnergiaEP.model.dao.entities.Tipousuario

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Tipousuario**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the bitacora_ingreso database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Tipousuario()

Method Summary

Methods	
Modifier and Type	Method and Description
Funcionario	addFuncionario (Funcionario funcionario)
java.util.List< Funcionario >	getFuncionarios ()
java.lang.Integer	getIdtipousuario ()
java.lang.String	getUsuario ()
Funcionario	removeFuncionario (Funcionario funcionario)
void	setFuncionarios (java.util.List< Funcionario > funcionarios)
void	setIdtipousuario (java.lang.Integer idtipousuario)
void	setUsuario (java.lang.String usuario)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Funcionario

java.lang.Object

└─ controlEnergiaEP.model.dao.entities.Funcionario

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Funcionario**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the bitacora_ingreso database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Funcionario()

Method Summary

Methods	
Modifier and Type	Method and Description
Evento	addEvento (Evento evento)
java.lang.String	getApellidos ()
java.lang.String	getCedula ()
java.lang.String	getClave ()
java.lang.String	getCorreo ()
java.lang.Boolean	getEstado ()
java.util.List< Evento >	getEventos ()
java.lang.String	getNombres ()
Tipousuario	getTipousuario ()
Evento	removeEvento (Evento evento)
void	setApellidos (java.lang.String apellidos)
void	setCedula (java.lang.String cedula)
void	setClave (java.lang.String clave)
void	setCorreo (java.lang.String correo)
void	setEstado (java.lang.Boolean estado)
void	setEventos (java.util.List< Evento > eventos)
void	setNombres (java.lang.String nombres)
void	setTipousuario (Tipousuario tipousuario)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Evento

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Evento

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Evento**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the bitacora_salida database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Evento()

Method Summary

Methods	
Modifier and Type	Method and Description
java.lang.String	getAccion()
java.lang.Integer	getCodigoevento()
java.lang.String	getDetalle()
java.util.Date	getFechaEvento()
Funcionario	getFuncionario()
java.lang.String	getIpcomputador()
void	setAccion() (java.lang.String accion)
void	setCodigoevento() (java.lang.Integer codigoevento)
void	setDetalle() (java.lang.String detalle)
void	setFechaEvento() (java.util.Date fechaEvento)
void	setFuncionario() (Funcionario funcionario)
void	setIpcomputador() (java.lang.String ipcomputador)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Modelo

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Modelo

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Modelo**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the controles database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Modelo()

Method Summary

Methods	
Modifier and Type	Method and Description
Raspberry	addRaspberry (Raspberry raspberry)
java.lang.String	getDescripcion ()
java.lang.Integer	getIdmodelo ()
java.lang.String	getNombre ()
java.lang.Integer	getPin1 ()
java.lang.Integer	getPin2 ()
java.lang.Integer	getPin3 ()
java.lang.Integer	getPin4 ()
java.util.List< Raspberry >	getRaspberries ()
Raspberry	removeRaspberry (Raspberry raspberry)
void	setDescripcion (java.lang.String descripcion)
void	setIdmodelo (java.lang.Integer idmodelo)
void	setNombre (java.lang.String nombre)
void	setPin1 (java.lang.Integer pin1)
void	setPin2 (java.lang.Integer pin2)
void	setPin3 (java.lang.Integer pin3)
void	setPin4 (java.lang.Integer pin4)
void	setRaspberries (java.util.List< Raspberry > raspberries)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Raspberry

java.lang.Object

└─ controlEnergiaEP.model.dao.entities.Raspberry

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Raspberry**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the lugares database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Raspberry()

Method Summary

Methods	
Modifier and Type	Method and Description
Dispositivo	addDispositivo (Dispositivo dispositivo)
java.lang.String	getDescripcion ()
java.util.List< Dispositivo >	getDispositivos ()
java.lang.Integer	getIdraspberry ()
java.lang.String	getIpraspberry ()
Modelo	getModelo ()
java.lang.String	getNombre ()
java.lang.String	getObservacion ()
java.lang.String	getUbicacion ()
Dispositivo	removeDispositivo (Dispositivo dispositivo)
void	setDescripcion (java.lang.String descripcion)
void	setDispositivos (java.util.List< Dispositivo > dispositivos)
void	setIdraspberry (java.lang.Integer idraspberry)
void	setIpraspberry (java.lang.String ipraspberry)
void	setModelo (Modelo modelo)
void	setNombre (java.lang.String nombre)
void	setObservacion (java.lang.String observacion)
void	setUbicacion (java.lang.String ubicacion)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Tipodispositivo

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Tipodispositivo

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Tipodispositivo**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the personas database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Tipodispositivo()

Method Summary

Methods	
Modifier and Type	Method and Description
Dispositivo	addDispositivo (Dispositivo dispositivo)
java.lang.Integer	getConsumo ()
java.util.List< Dispositivo >	getDispositivos ()
java.lang.Integer	getIdtipodispositivo ()
java.lang.String	getNombre ()
Dispositivo	removeDispositivo (Dispositivo dispositivo)
void	setConsumo (java.lang.Integer consumo)
void	setDispositivos (java.util.List< Dispositivo > dispositivos)
void	setIdtipodispositivo (java.lang.Integer idtipodispositivo)
void	setNombre (java.lang.String nombre)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Dispositivo

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Dispositivo

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Dispositivo**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the transporte database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Dispositivo()

Method Summary

Methods	
Modifier and Type	Method and Description
Actividad	addActividad (Actividad actividad)
java.util.List< Actividad >	getActividades ()
java.lang.String	getDescripcion ()
java.lang.Boolean	getEstado ()
java.lang.Integer	getIddispositivo ()
java.lang.String	getObservacion ()
java.lang.Integer	getPuerto ()
Raspberry	getRaspberry ()
Tipodispositivo	getTipodispositivo ()
java.lang.String	getUbicacion ()
Actividad	removeActividad (Actividad actividad)
void	setActividades (java.util.List< Actividad > actividades)
void	setDescripcion (java.lang.String descripcion)
void	setEstado (java.lang.Boolean estado)
void	setIddispositivo (java.lang.Integer iddispositivo)
void	setObservacion (java.lang.String observacion)
void	setPuerto (java.lang.Integer puerto)
void	setRaspberry (Raspberry raspberry)
void	setTipodispositivo (Tipodispositivo tipodispositivo)
void	setUbicacion (java.lang.String ubicacion)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Actividad

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Actividad

All Implemented Interfaces:

java.io.Serializable

@Entity

public class **Actividad**

extends java.lang.Object

implements java.io.Serializable

The persistent class for the turismo database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Actividad()

Method Summary

Methods	
Modifier and Type	Method and Description
Registro	addRegistro (Registro registro)
Dispositivo	getDispositivo ()
java.lang.Boolean	getDomingo ()
java.lang.Boolean	getEstado ()
java.util.Date	getFechaActual ()
java.util.Date	getFechaFin ()
java.util.Date	getFechaInicio ()
java.util.Date	getHoraDesde ()
java.util.Date	getHoraHasta ()
java.lang.Integer	getIdactividad ()
java.lang.Boolean	getJueves ()
java.lang.Boolean	getLunes ()
java.lang.Boolean	getMartes ()
java.lang.Boolean	getMiercoles ()
java.lang.String	getNombre ()
java.lang.String	getObservacion ()
java.util.List< Registro >	getRegistros ()
java.lang.Boolean	getSabado ()
java.lang.Boolean	getViernes ()
Registro	removeRegistro (Registro registro)
void	setDispositivo (Dispositivo dispositivo)
void	setDomingo (java.lang.Boolean domingo)
void	setEstado (java.lang.Boolean estado)
void	setFechaActual (java.util.Date fechaActual)
void	setFechaFin (java.util.Date fechaFin)
void	setFechaInicio (java.util.Date fechaInicio)

void	setHoraDesde (java.util.Date horaDesde)
void	setHoraHasta (java.util.Date horaHasta)
void	setIdactividad (java.lang.Integer idactividad)
void	setJueves (java.lang.Boolean jueves)
void	setLunes (java.lang.Boolean lunes)
void	setMartes (java.lang.Boolean martes)
void	setMiercoles (java.lang.Boolean miercoles)
void	setNombre (java.lang.String nombre)
void	setObservacion (java.lang.String observacion)
void	setRegistros (java.util.List<Registro> registros)
void	setSabado (java.lang.Boolean sabado)
void	setViernes (java.lang.Boolean viernes)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Class Registro

java.lang.Object

└ controlEnergiaEP.model.dao.entities.Registro

All Implemented Interfaces:

java.io.Serializable

@Entity
 public class **Registro**
 extends java.lang.Object
 implements java.io.Serializable

The persistent class for the usuarios database table.

See Also:

[Serialized Form](#)

Constructor Summary

Constructors
Constructor and Description
Registro()

Method Summary

Methods	
Modifier and Type	Method and Description
Actividad	<u>getActividad()</u>
java.sql.Timestamp	<u>getFin()</u>
java.lang.Integer	<u>getIdregistro()</u>
java.sql.Timestamp	<u>getInicio()</u>
java.lang.String	<u>getNombre()</u>
java.math.BigDecimal	<u>getTotalH()</u>
void	<u>setActividad(Actividad actividad)</u>
void	<u>setFin(java.sql.Timestamp fin)</u>
void	<u>setIdregistro(java.lang.Integer idregistro)</u>
void	<u>setInicio(java.sql.Timestamp inicio)</u>
void	<u>setNombre(java.lang.String nombre)</u>
void	<u>setTotalH(java.math.BigDecimal totalH)</u>

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Aplicación Web

Package controlEnergiaEP.model.manager

Class ManagerDAO

java.lang.Object

└ controlEnergiaEP.model.manager.ManagerDAO

```
public class ManagerDAO
  extends java.lang.Object
```

Objeto que encapsula la lógica básica de acceso a datos mediante JPA. Maneja el patrón de diseño singleton para administrar los componentes EntityManagerFactory y EntityManager.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

Manager DAO

```
public ManagerDAO()
```

Constructor de la clase ManagerDAO. Se encarga de crear los objetos factory y entity manager utilizando el patrón de diseño singleton.

Method Detail

mostrarLog

```
public void mostrarLog(java.lang.Class clase, java.lang.String nombreMetodo, java.lang.String mensaje)
```

Método básico para mostrar mensajes de depuración.

Parameters:

- clase - Información de la clase (Class) para generar el mensaje de depuración.
- nombreMetodo - Método que genera el mensaje de depuración.
- mensaje - El mensaje a desplegar.

findAll

```
public java.util.List findAll(java.lang.Class clase, java.lang.String orderBy)
```

finder Genérico que devuelve todas las entidades de una tabla.

Parameters:

- clase - La clase que se desea consultar. Por ejemplo:
 - Usuario.class

orderBy - Expresión que indica la propiedad de la entidad por la que se desea ordenar la consulta.

Debe utilizar el alias "o" para nombrar a la(s) propiedad(es) por la que se va a ordenar. Por ejemplo:

- o.nombre
- o.codigo,
- o.nombre

Puede aceptar null o una cadena vacía, en este caso no ordenará el resultado.

Returns:

Listado resultante.

findAll

```
public java.util.List findAll(java.lang.Class clase)
```

finder Genérico que devuelve todas las entidades de una tabla.

Parameters:

- clase - La clase que se desea consultar. Por ejemplo:
 - Usuario.class

Returns:

Listado resultante.

findWhere

```
public java.util.List findWhere(java.lang.Class clase, java.lang.String pClausulaWhere, java.lang.String pOrderBy)
```

Finder genérico que permite aplicar cláusulas where y order by.

Parameters:

- clase - La entidad sobre la que se desea consultar. Ej: Usuario.class
- pClausulaWhere - Clausula where de tipo JPQL (sin la palabra reservada WHERE).

Ejemplo:

- o.nombre='Antonio'
- o.nombre='Antonio' and o.telefono='0444-434'
- o.nombre like 'Ant%'

pOrderBy - Clausula order by de tipo JPQL (sin la palabra reservada ORDER BY).

Puede ser null para no ordenar. Por ejemplo:
o.nombre

o.codigo, o.nombre
Tanto para la clausula where como order by debe utilizarse el alias de entidad "o".

Returns:
Listado resultante.

findById

public java.lang.Object findById(java.lang.Class clase, java.lang.Object pID) throws java.lang.Exception
Finder genérico para buscar un objeto específico.

Parameters:
clase - La clase sobre la que se desea consultar, ejemplo: Usuario.class
pID - Identificador (la clave primaria) que permitirá la búsqueda.

Returns:
El objeto solicitado (si existiera).

Throws:
java.lang.Exception

verificar

public java.lang.Boolean verificar(java.lang.String pObjeto)
Almacena un objeto (persistencia).

Parameters:
pObjeto - El objeto a insertar.

Throws:
java.lang.Exception

insertar

public void insertar(java.lang.Object pObjeto) throws java.lang.Exception

Throws:
java.lang.Exception

eliminar

public void eliminar(java.lang.Class clase, java.lang.Object pID) throws java.lang.Exception
Elimina un objeto de la persistencia.

Parameters:
clase - La clase correspondiente al objeto que se desea eliminar.
pID - El identificador del objeto que se desea eliminar.

Throws:
java.lang.Exception

actualizar

public void actualizar(java.lang.Object pObjeto) throws java.lang.Exception
Actualiza la información de un objeto en la persistencia.

Parameters:
pObjeto - Objeto que contiene la información que se debe actualizar.

Throws:
java.lang.Exception

Class ManagerActividad

java.lang.Object

└ controlEnergiaEP.model.manager.Actividad

```
public class ManagerActividad  
extends java.lang.Object
```

Clase que implementa la lógica del calendario del sistema (CRUD) y búsqueda por fechas de las actividades creadas.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

ManagerAuditoria

```
public ManagerAuditoria()
```

Method Detail

crearEvento

```
public void crearEvento(java.lang.String cedula, java.lang.String accion, java.lang.String detalle) throws  
java.lang.Exception
```

Almacena la información de un evento en la tabla de auditoría.

Parameters:

cedula - Código del usuario que genero el evento.
accion - Nombre del método que genero el evento.
detalle - Informacion detallada del evento.

Throws:

java.lang.Exception

findEventosFechas

```
public java.util.List<Evento> findEventosFechas(java.util.Date fechaInicio, java.util.Date fechaFinal)  
throws java.lang.Exception
```

Busca en los eventos de auditoría por fechas.

Parameters:

fechaInicio - fecha de inicio de la búsqueda de los eventos.
fechaFinal - fecha de fin de la búsqueda de los eventos.

Returns:

listado de eventos de auditoría.

Throws:

java.lang.Exception

findAllEventos

```
public java.util.List<Evento> findAllEventos()
```

Class ManagerCalendario

java.lang.Object

└─ controlEnergiaEP.model.manager.Calendario

```
public class ManagerCalendario  
extends java.lang.Object
```

Clase que implementa la lógica del calendario del sistema (CRUD) y búsqueda por fechas de las actividades creadas.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

ManagerCalendario

```
public ManagerCalendario()
```

Method Detail

findAllActividad

```
public java.util.List<Actividad> findAllActividad()
```

Metodo finder para consulta de una actividad. Hace uso del componente `ManagerDAO` de la capa model.

Returns:

listado de las actividades.

findActividadById

```
public Actividad findActividadById(java.lang.Integer idActividad) throws java.lang.Exception
```

Metodo finder para consulta de la actividad. Hace uso del componente `ManagerDAO` de la capa model.

Parameters:

idActividad - código de la actividad que se desea buscar.

Returns:

la actividad encontrada.

Throws:

java.lang.Exception

insertarActividad

```
public void insertarActividad(java.lang.String cedula, Actividad a) throws java.lang.Exception
```

Guarda una nueva actividad en la base de datos. Hace uso del componente `ManagerDAO` de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que inserta la actividad.

a - La nueva actividad.

Throws:

java.lang.Exception

actualizarActividad

```
public void actualizarActividad(Actividad actividad) throws java.lang.Exception
```

Actualiza la información de una actividad en la base de datos. Hace uso del componente `ManagerDAO` de la capa model.

Parameters:

`actividad` - Los datos de la actividad que se desea actualizar.

Throws:

`java.lang.Exception`

findMostrarListaActividad

```
public java.util.List<Actividad> findMostrarListaActividad(java.util.Date fechaIni, java.util.Date fechaFi)
throws java.lang.Exception
```

Muestra las actividades insertadas en el rango de `fechaInicio` y `fechaFinal`

Parameters:

`fechaIni` - fecha de inicio de la búsqueda.

`fechaFi` - fecha final de la búsqueda.

Returns:

 listado de actividades ingresadas.

Throws:

`java.lang.Exception`

Class ManagerControlEnergia

`java.lang.Object`

└─ `controlEnergiaEP.model.manager.ManagerControlEnergia`

```
public class ManagerControlEnergia
extends java.lang.Object
```

Clase que implementa las reglas de negocio de las entidades (Dispositivo, Funcionario, Modelo, Parametro, Rasperry, Tipodispositivo, Tipousuario) realizando el (CRUD).

Author:

Jenny Rea - YACHAY-EP

Constructor Detail

ManagerControlEnergia

```
public ManagerControlEnergia()
```

Method Detail

findAllModelo

```
public java.util.List<Modelo> findAllModelo()
```

Método finder para consulta de Modelo. Hace uso del componente `ManagerDAO` de la capa model.

Returns:

 listado de modelos ordenados por nombre.

findModeloById

```
public Modelo findModeloById(java.lang.Integer idModelo) throws java.lang.Exception
```

Método finder para consulta de modelo. Hace uso del componente `ManagerDAO` de la capa model.

Parameters:

idModelo - código del modelo que se desea buscar.

Returns:

el modelo encontrado.

Throws:

`java.lang.Exception`

insertarModelo

```
public void insertarModelo(java.lang.String cedula, Modelo m) throws java.lang.Exception
```

Guarda un nuevo modelo en la base de datos. Hace uso del componente `ManagerDAO` de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que inserta el modelo.

m - El nuevo modelo.

Throws:

`java.lang.Exception`

eliminarModelo

```
public void eliminarModelo(java.lang.String cedula, java.lang.Integer idModelo) throws java.lang.Exception
```

Borra de la base de datos un modelo específico. Hace uso del componente `ManagerDAO` de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que elimina el modelo.

idModelo - el código del modelo que se desea eliminar.

Throws:

`java.lang.Exception`

actualizarModelo

```
public void actualizarModelo(java.lang.String cedula, Modelo modelo) throws java.lang.Exception
```

Actualiza la información de un modelo en la base de datos. Hace uso del componente `ManagerDAO` de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que actualiza el modelo.

modelo - Los datos del modelo que se desea actualizar.

Throws:

`java.lang.Exception`

findAllRaspberry

```
public java.util.List<Raspberry> findAllRaspberry()
```

Método finder para consulta de Raspberry. Hace uso del componente `ManagerDAO` de la capa model.

Returns:

listado de Raspberries ordenados por nombre.

findRaspberryById

public Raspberry findRaspberryById(java.lang.Integer idRaspberry) throws java.lang.Exception

Método finder para consulta de raspberries. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idRaspberry - código del raspberry que se desea buscar.

Returns:

el raspberry encontrado.

Throws:

java.lang.Exception

insertarRaspberry

public void insertarRaspberry(java.lang.String cedula, Raspberry r) throws java.lang.Exception

Guarda un nuevo raspberry en la base de datos. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que inserta el raspberry.

r - El nuevo raspberry.

Throws:

java.lang.Exception

eliminarRaspberry

public void eliminarRaspberry(java.lang.String cedula, java.lang.Integer idRaspberry) throws java.lang.Exception

Borra de la base de datos un raspberry específico. Hace uso del componente {controlEnergiaEP.@link model.manager.ManagerDAO ManagerDAO} de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que elimina el raspberry.

idRaspberry - el código del raspberry que se desea eliminar.

Throws:

java.lang.Exception

actualizarRaspberry

public void actualizarRaspberry(java.lang.String cedula,Raspberry raspberry) throws java.lang.Exception

Actualiza la información de un raspberry en la base de datos. Hace uso del componente {controlEnergiaEP.@link model.manager.ManagerDAO ManagerDAO} de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que actualiza el raspberry.

raspberry - Los datos del raspberry que se desea actualizar.

Throws:

java.lang.Exception

findAllTipoDispositivo

public java.util.List<Tipodispositivo> findAllTipoDispositivo()

Método finder para consulta de tipoDispositivo. Hace uso del componente ManagerDAO de la capa model.

Returns:
listado de tipoDispositivo ordenados por nombre.

findTipoDispositivoById

public Tipodispositivo findTipoDispositivoById(java.lang.Integer idtipodispositivo) throws java.lang.Exception

Método finder para consulta de tipoDispositivo. Hace uso del componente ManagerDAO de la capa model.

Parameters:
idtipodispositivo - código del tipoDispositivo que se desea buscar.

Returns:
el tipoDispositivo encontrado.

Throws:
java.lang.Exception

insertarTipoDispositivo

public void insertarTipoDispositivo(java.lang.String cedula, Tipodispositivo td) throws java.lang.Exception

Guarda un nuevo tipoDispositivo en la base de datos. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:
cedula - cédula del funcionario que inserta el tipo de dispositivo.
td - El nuevo tipoDispositivo.

Throws:
java.lang.Exception

eliminarTipoDispositivo

public void eliminarTipoDispositivo(java.lang.String cedula, java.lang.Integer idtipodispositivo) throws java.lang.Exception

Borra de la base de datos un tipoDispositivo específico. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:
cedula - cédula del funcionario que elimina el tipo de dispositivo.
idtipodispositivo - el código del tipoDispositivo que se desea eliminar.

Throws:
java.lang.Exception

actualizarTipoDispositivo

public void actualizarTipoDispositivo(java.lang.String cedula, Tipodispositivo tipoDispositivo) throws java.lang.Exception

Actualiza la información de un tipoDispositivo en la base de datos. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:
cedula - cédula del funcionario que actualiza el tipo de dispositivo.
tipoDispositivo - Los datos del tipoDispositivo que se desea actualizar.

Throws:
java.lang.Exception

findAllDispositivo

```
public java.util.List<Dispositivo> findAllDispositivo()
```

Método finder para consulta del dispositivo. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de dispositivos ordenados por nombre.

findDispositivoById

```
public Dispositivo findDispositivoById(java.lang.Integer idDispositivo) throws java.lang.Exception
```

Método finder para consulta del dispositivo. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idDispositivo - código del dispositivo que se desea buscar.

Returns:

el dispositivo encontrado.

Throws:

java.lang.Exception

insertarDispositivo

```
public void insertarDispositivo(java.lang.String cedula, Dispositivo d) throws java.lang.Exception
```

Guarda un nuevo dispositivo en la base de datos y se registra en el método de Auditoría. Hace uso del componente ManagerDAO de la capa model.

Parameters:

cedula - cédula del funcionario que inserta el dispositivo.

d - El nuevo dispositivo.

Throws:

java.lang.Exception

eliminarDispositivo

```
public void eliminarDispositivo(java.lang.String cedula, java.lang.Integer idDispositivo) throws java.lang.Exception
```

Borra de la base de datos un dispositivo. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que elimina el dispositivo.

idDispositivo - el código del dispositivo que se desea eliminar.

Throws:

java.lang.Exception

actualizarDispositivo

```
public void actualizarDispositivo(java.lang.String cedula, Dispositivo dispositivo) throws java.lang.Exception
```

Actualiza la información de un dispositivo en la base de datos. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que actualiza el dispositivo.

dispositivo - Los datos del dispositivo que se desea actualizar.

Throws:

java.lang.Exception

findAllTipoUsuario

```
public java.util.List<Tipousuario> findAllTipoUsuario()
```

Método finder para consulta del tipo de usuario. Hace uso del componente ManagerDAO de la capa model.

Returns:
listado de tipos de usuarios ordenados por usuario).

findTipoUsuarioByld

```
public Tipousuario findTipoUsuarioByld(java.lang.Integer idTipoUsuario) throws java.lang.Exception
```

Método finder para consulta del tipo de usuario. Hace uso del componente ManagerDAO de la capa model.

Parameters:
idTipoUsuario - código del tipo de usuario que se desea buscar.

Returns:
el tipo de usuario encontrado.

Throws:
java.lang.Exception

findAllFuncionario

```
public java.util.List<Funcionario> findAllFuncionario()
```

Método finder para consulta del funcionario. Hace uso del componente ManagerDAO de la capa model.

Returns:
listado de funcionarios ordenados por apellidos).

findFuncionarioByld

```
public Funcionario findFuncionarioByld(java.lang.String cedula) throws java.lang.Exception
```

Método finder para consulta de funcionarios. Hace uso del componente ManagerDAO de la capa model.

Parameters:
cedula - código del funcionario que se desea buscar.

Returns:
el funcionario encontrado.

Throws:
java.lang.Exception

insertarFuncionario

```
public void insertarFuncionario(java.lang.String cedula, Funcionario f) throws java.lang.Exception
```

Guarda un nuevo funcionario en la base de datos. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:
cedula - cédula del funcionario que inserta el nuevo usuario.
f - El nuevo funcionario.

Throws:
java.lang.Exception

actualizarFuncionario

```
public void actualizarFuncionario(Funcionario funcionario) throws java.lang.Exception
```

Actualiza la información de un funcionario en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

funcionario - Los datos del funcionario que se desea actualizar.

Throws:

java.lang.Exception

validadorDeCedula

public boolean validadorDeCedula(java.lang.String cedula)

Método para efectuar la validación de la cédula de identidad de las personas a ingresar en el sistema.

Parameters:

cedula - cédula a comprobar.

Returns:

boolean true o false.

findAllPermiso

public java.util.List<Parametro> findAllPermiso()

Método finder para consulta de parametros. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de Parámetros ordenados por nombre.

findParametroById

public Parametro findParametroById(java.lang.String nombre) throws java.lang.Exception

Método finder para consulta de parámetros. Hace uso del componente ManagerDAO de la capa model.

Parameters:

nombre - código del parámetro que se desea buscar.

Returns:

el parámetro encontrado.

Throws:

java.lang.Exception

insertarParametro

public void insertarParametro(java.lang.String cedula, Parametro p) throws java.lang.Exception

Guarda un nuevo parámetro en la base de datos. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que inserta el parámetro.

p - El nuevo parámetro.

Throws:

java.lang.Exception

eliminarParametro

```
public void eliminarParametro(java.lang.String cedula, java.lang.String nombre) throws  
java.lang.Exception
```

Borra de la base de datos un parametro especifico. Hace uso del componente ManagerDAO de la capa model. Además se inserta como evento en la tabla de auditoría.

Parameters:

cedula - cédula del funcionario que elimina el parámetro.
nombre - el código del parámetro que se desea eliminar.

Throws:

java.lang.Exception

actualizarParametro

```
public void actualizarParametro(java.lang.String cedula, Parametro parametro) throws  
java.lang.Exception
```

Actualiza la información de un parámetro en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

cedula - cédula del funcionario que actualiza el parámetro.
parametro - Los datos del parámetro que se desea actualizar.

Throws:

java.lang.Exception

encriptar

```
public static java.lang.String encriptar(java.lang.String cadena)
```

Encripta un mensaje de texto mediante algoritmo de resumen de mensaje.

Parameters:

cadena - texto a encriptar.

Returns:

mensaje encriptado.

Class ManagerReportes

java.lang.Object

└ controlEnergiaEP.model.manager.ManagerReportes

```
public class ManagerReportes
```

```
extends java.lang.Object
```

Clase que implementa la lógica de reportes estadísticos del sistema.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

ManagerReportes

```
public ManagerReportes()
```

Method Detail

findSumaConsumo

```
public java.util.List<java.lang.Object[]> findSumaConsumo(java.lang.Class clase,
java.util.Date fechalnicio, java.util.Date fechaFinal) throws java.lang.Exception
```

Muestra el total de horas de cada dispositivo ingresado, agrupado por el tipo de dispositivo y sumando las mismas por medio de la sentencia sum.

Parameters:

clase - clase en la que se va a realizar la búsqueda
fechalnicio - fecha de inicio de la búsqueda.
fechaFinal - fecha final de la búsqueda.

Returns:

listado de registros agrupados por tipo de dispositivo con su suma respectiva.

Throws:

java.lang.Exception

Package controlEnergiaEP.controller

Class BeanAplicacion

java.lang.Object

└─ controlEnergiaEP.controller.BeanAplicacion

```
public class BeanAplicacion
extends java.lang.Object
```

Clase que permite ingresar al Sistema de Control de Energía.

Author:

Jenny Rea - YACHAY EP.

Constructor Summary

Constructors
Constructor and Description
BeanAplicacion()

Method Summary

Methods	
Modifier and Type	Method and Description
boolean	isInicioSesion()
void	setInicioSesion (boolean inicioSesion)

Class BeanCalendario

java.lang.Object

└ controlEnergiaEP.controller.BeanCalendario

```
public class BeanCalendario  
extends java.lang.Object
```

Clase que genera las actividades del sistema para el encendido y apagado de un dispositivo.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

BeanCalendario

```
public BeanCalendario()
```

Method Detail

mostrarEstado

```
public void mostrarEstado()
```

Método informativo que muestra si el dispositivo tiene una actividad a realizar.

onFechaInicioSelect

```
public void onFechaInicioSelect(org.primefaces.event.SelectEvent event)
```

identificarDia

```
public void identificarDia()
```

Método que permite identificar los días que se encuentran en el rango escogido de la FechaInicio y FechaFin; para deshabilitar los días que no pertenecen al mismo se hace uso de los estados booleanos de los días respectivamente. En el caso de que el rango sea mayor o igual a 7 días se habilitarán todos los campos de los días.

verificarDispositivoActividad

```
public boolean verificarDispositivoActividad()
```

Método que verifica las actividades activas que tiene un dispositivo específico; la nueva actividad a crear no puede abarcar las fechas de las actividades antes ingresadas.

Returns:

estado booleano true(cuando no existen actividades en las fechas escogidas) o false(si existe actividades en dichas fechas).

Throws:

java.lang.Exception

getListaBusquedaActividad

```
public java.util.List<Actividad> getListaBusquedaActividad()
```

Muestra por defecto la lista de actividades creadas en la última semana; en el caso de querer obtener la lista en otro rango de fechas escoja las mismas en la página web.

Returns:

la lista de actividades creadas en las fechas seleccionadas.

Class BeanDispositivo

java.lang.Object

└ controlEnergiaEP.controller.BeanDispositivo

```
public class BeanDispositivo
```

```
extends java.lang.Object
```

Clase que permite efectuar el CRUD de los dispositivos que se usan en el sistema (ejemplo: televisión).

Author:

Jenny Rea - YACHAY EP.

Constructor Detail**BeanDispositivo**

```
public BeanDispositivo()
```

Method Detail**getListaRaspberrySI**

```
public java.util.List<Raspberry> getListaRaspberrySI()
```

Devuelve un listado de componentes SelectListItem a partir de un listado de Raspberry.

Returns:

listado de SelectListItem de un raspberry.

getListaTipoDispositivoSI

```
public java.util.List<Tipodispositivo> getListaTipoDispositivoSI()
```

Devuelve un listado de componentes SelectListItem a partir de un listado de Tipodispositivo.

Returns:

listado de SelectListItem de un tipoDispositivo.

buscarPuerto

```
public boolean buscarPuerto()
```

Método que permite ver la disponibilidad de un puerto o pin de un equipo raspberry específico; generando el mensaje correspondiente de ser necesario.

Returns:

un booleano true o false.

Class BeanEventos

java.lang.Object

└ controlEnergiaEP.controller.BeanEventos

```
public class BeanEventos
```

```
extends java.lang.Object
```

Clase que registra las acciones que se llevan acabo en el sistema para mostrar en una lista de eventos mediante un rango de fechas.

Author:

Jenny Rea - YACHAY EP.

Constructor Summary

Constructors
Constructor and Description
BeanEventos()

Method Summary

Methods	
Modifier and Type	Method and Description
java.lang.String	<u>actionBuscar()</u>
java.lang.String	<u>getAccion()</u>
java.lang.String	<u>getDetalle()</u>
java.lang.Boolean	<u>getFechaEstado()</u>
java.util.Date	<u>getFechaEvento()</u>
java.util.Date	<u>getFechaFin()</u>
java.util.Date	<u>getFechaInicio()</u>
java.util.Date	<u>getFechaMaxima()</u>
java.util.Date	<u>getFechaMin()</u>
java.lang.String	<u>getIdUsuario()</u>
java.lang.String	<u>getIpComputador()</u>
java.util.List<Evento>	<u>getListaEventos()</u>
java.util.List<Evento>	<u>getMostrarEventos()</u>
void	<u>onFechaInicioSelect</u> (org.primefaces.event.SelectEvent event)
void	<u>setAccion</u> (java.lang.String accion)
void	<u>setDetalle</u> (java.lang.String detalle)
void	<u>setFechaEstado</u> (java.lang.Boolean fechaEstado)
void	<u>setFechaEvento</u> (java.util.Date fechaEvento)
void	<u>setFechaFin</u> (java.util.Date fechaFin)
void	<u>setFechaInicio</u> (java.util.Date fechaInicio)
void	<u>setFechaMaxima</u> (java.util.Date fechaMaxima)
void	<u>setFechaMin</u> (java.util.Date fechaMin)
void	<u>setIdUsuario</u> (java.lang.String idUsuario)
void	<u>setIpComputador</u> (java.lang.String ipComputador)
void	<u>setListaEventos</u> (java.util.List<Evento> listaEventos)

Class BeanFuncionario

java.lang.Object

└ controlEnergiaEP.controller.BeanFuncionario

```
public class BeanFuncionario  
extends java.lang.Object
```

Clase que permite crear, actualizar, visualizar, deshabilitar a un funcionario de la empresa.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

BeanFuncionario

```
public BeanFuncionario()
```

Method Detail

getListaTipoUsuarioSI

```
public java.util.List<Tipousuario> getListaTipoUsuarioSI()
```

Devuelve un listado de componentes SelectListItem a partir de un listado de Tipousuario.

Returns:

listado de SelectListItem de tipo de usuario.

reset

```
public void reset(Funcionario f)
```

Método que permite resetear la contraseña de un funcionario específico. La contraseña se actualiza con la contraseña por defecto ingresada en la página de parámetros (clave encriptada). Adicional a lo mencionado si el parámetro (AC) tiene como valor (SI) envía un mail al correo del funcionario que se ha modificado la contraseña.

Parameters:

f - funcionario que se desea resetear la contraseña.

desactivar

```
public void desactivar(Funcionario f)
```

Método que permite deshabilitar o habilitar a un funcionario específico del uso del sistema. Se actualiza el atributo (estado).

Parameters:

f - funcionario que se desea desactivar.

Class BeanLogin

java.lang.Object

└ controlEnergiaEP.controller.BeanLogin

```
public class BeanLogin
extends java.lang.Object
```

Clase que permite ingresar al Sistema de Control de Energía.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

BeanLogin

```
public BeanLogin()
```

Method Detail

comprobarIngreso

```
public java.lang.String comprobarIngreso()
```

Método que permite al usuario acceder al sistema. (Para mayor seguridad las contraseñas se encuentran encriptadas (MD5)).

Returns:

Retorna el tipo de usuario (Administrador o Usuario). En el caso de ser Administrador se redirecciona a la página de index; de ser Usuario se visualiza la página de Calendario del sistema. Si el Usuario ingresa al sistema por primera vez se muestra la página de CambioClave.

Throws:

java.lang.Exception - Cuando no coincide la clave proporcionada o si ocurrió un error con la consulta a la base de datos.

salirSistema

```
public java.lang.String salirSistema()
```

Finaliza la sesión web del usuario.

Returns:

la página principal del sistema (Login).

Class BeanModelo

java.lang.Object

└ controlEnergiaEP.controller.BeanModelo

```
public class BeanModelo  
extends java.lang.Object
```

Clase que permite generar el CRUD de los modelos de Raspberrys (El modelo a usar es el B).

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

BeanModelo

```
public BeanModelo()
```

Method Detail

comparar

```
public java.lang.Boolean comparar()
```

Método que compara los pines ingresados en la página web para evitar que se repitan los valores de los mismos.

Returns:

valor boolean true (no existen valores repetidos) o false.

Class BeanParametros

java.lang.Object

└ controlEnergiaEP.controller.BeanParametros

```
public class BeanParametros  
extends java.lang.Object
```

Clase que permite generar el CRUD de los parámetros necesarios en el sistema.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

BeanParametros

```
public BeanParametros()
```

Method Detail

actionActualizarClavePrivada

```
public void actionActualizarClavePrivada()
```

Método que permite actualizar la clave por defecto mediante un cuadro diálogo (Para mayor seguridad esta clave se encripta con el método MD5).

Class BeanRaspberry

java.lang.Object

└ controlEnergiaEP.controller.BeanRaspberry

```
public class BeanRaspberry
extends java.lang.Object
```

Clase que permite crear el CRUD de un Raspberry Pi.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

BeanRaspberry

```
public BeanRaspberry()
```

getListaModeloSI

```
public java.util.List<Modelo> getListaModeloSI()
```

Devuelve un listado de componentes SelectItem a partir de un listado de Modelo.

Returns:

listado de SelectItems de un modelo.

Class BeanReporte

java.lang.Object

└ controlEnergia.controller.BeanReporte

All Implemented Interfaces:

java.io.Serializable

```
public class BeanReporte
extends java.lang.Object
implements java.io.Serializable
```

Clase que permite generar los reportes estadísticos del sistema mediante la clase BarChartModel (gráficos estadísticos verticales de Primafaces).

Author:

Jenny Rea - YACHAY EP.

See Also:

[Serialized Form](#)

Constructor Detail

BeanReporte

```
public BeanReporte()
```

Method Detail

```
init  
@PostConstruct  
public void init()
```

```
getBarDispositivo  
public org.primefaces.model.chart.BarChartModel getBarDispositivo()
```

```
getBarDispositivoWatts  
public org.primefaces.model.chart.BarChartModel getBarDispositivoWatts()
```

```
initBarModelDispositivoConsumoHoras  
public org.primefaces.model.chart.BarChartModel initBarModelDispositivoConsumoHoras()  
Elabora el gráfico estadístico de las horas consumidas por todos los dispositivos;  
Returns:  
    BarChartModel dispositivo
```

```
getListaDispositivosConsumoHoras  
public java.util.List<java.lang.Object[]> getListaDispositivosConsumoHoras()  
Muestra una lista del consumo eléctrico en horas de los dispositivos implementados en el sistema;  
determinado por el rango de fechas escogidas en la página web.  
Returns:  
    lista de Dispositivos con su consumo respectivamente
```

```
initBarModelDispositivoConsumoKWTTTS  
public org.primefaces.model.chart.BarChartModel initBarModelDispositivoConsumoKWTTTS()  
Elabora el gráfico estadístico del consumo de todos los dispositivos en KWtts. *  
Returns:  
    BarChartModel dispositivo
```

```
getListaDispositivoConsumoKWTTTS  
public java.util.List<java.lang.Object[]> getListaDispositivoConsumoKWTTTS()  
Muestra una lista del consumo eléctrico en KWTTTS de los dispositivos implementados en el  
sistema; determinado por el rango de fechas escogidas en la página web.  
Returns:  
    lista de Dispositivos con su consumo en KWTTTS respectivamente.
```

```
createBarModels1  
public java.lang.String createBarModels1()
```

createBarModelConsumo

```
public void createBarModelConsumo()
```

Crea los detalles del gráfico estadístico del consumo en horas. La variable "mayor" es el número más alto de los valores del gráfico; a éste se añade 50 puntos más de altura para que se visualice mejor las etiquetas del gráfico.

createBarModelConsumoWatts

```
public void createBarModelConsumoWatts()
```

Crea los detalles del gráfico estadístico del consumo en KWTS. La variable "mayor" es el número más alto de los valores del gráfico; a éste se añade 50 puntos más de altura para que se visualice mejor las etiquetas del gráfico.

Class BeanTipoDispositivo

```
java.lang.Object
```

```
└─ controlEnergiaEP.controller.BeanTipoDispositivo
```

```
public class BeanTipoDispositivo
```

```
extends java.lang.Object
```

Clase que permite generar el CRUD de los Tipos de Dispositivos del sistema.

Author:

Jenny Rea - YACHAY EP.

Constructor Summary

Constructors
Constructor and Description
<u>BeanTipoDispositivo()</u>

Method Summary

Methods	
Modifier and Type	Method and Description
java.lang.String	<u>actionActualizarTipoDispositivo()</u>
java.lang.String	<u>actionCargarTipoDispositivo(Tipodispositivo tipoDispositivo)</u>
java.lang.String	<u>actionEliminarTipoDispositivo(Tipodispositivo tipoDispositivo)</u>
java.lang.String	<u>actionInsertarTipoDispositivo()</u>
java.lang.String	<u>cancelarTipoDispositivo()</u>
BeanLogin	<u>getBeanLogin()</u>
java.lang.String	<u>getCedula()</u>
java.lang.Integer	<u>getConsumo()</u>
java.lang.Integer	<u>getIdtipodispositivo()</u>
java.util.List<Tipodispositivo>	<u>getListTipoDispositivo()</u>
java.lang.String	<u>getNombre()</u>
void	<u>setBeanLogin(BeanLogin beanLogin)</u>
void	<u>setCedula(java.lang.String cedula)</u>
void	<u>setConsumo(java.lang.Integer consumo)</u>
void	<u>setIdtipodispositivo(java.lang.Integer idtipodispositivo)</u>
void	<u>setNombre(java.lang.String nombre)</u>

Class ConexionRaspberry

java.lang.Object

└─ controlEnergiaEP.controller.ConexionRaspberry

```
public class BeanTurismo
extends java.lang.Object
```

Clase cliente/servidor que permite conectarse con el equipo Raspberry.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

ConexionRaspberry

```
public ConexionRaspberry()
```

Method Detail

accion

```
public void accion(java.lang.String cadena, java.lang.String ip) throws java.lang.Exception
```

Método que permite enviar datos al equipo Raspberry usando Sockets.

Parameters:

cadena - estado del dispositivo('encender/apagar'), pin al que se encuentra conectado el mismo.

ip - ip del Raspberry con el que se va a conectar.

Throws:

java.lang.Exception

Class MailUtil

java.lang.Object

└─ controlEnergiaEP.controller.MailUtil

```
public class MailUtil
extends java.lang.Object
```

Clase que permite enviar emails usando la librería JavaMail y el protocolo TLS.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

MailUtil

```
public MailUtil()
```


Method Detail

inicializar

public javax.mail.Session inicializar() throws java.lang.Exception
 Método que permite cargar los datos para iniciar una sesión con el servidor de correo electrónico (GMAIL).

Returns:
 la sesión con los datos correspondientes.

Throws:
 java.lang.Exception

enviarCorreo

public void enviarCorreo(java.lang.String direccionPara, java.lang.String asunto, java.lang.String contenido, java.lang.String nota) throws java.lang.Exception

Método que crea el mail con los mensajes respectivos y envía el mismo.

Parameters:
 direccionPara - dirección de correo electrónico a enviar el mail.
 asunto - razón por el cual se envía el mensaje.
 contenido - texto que va en el mail.
 nota - sugerencia o posdata del mail.

Throws:
 java.lang.Exception

Class JSFUtil

java.lang.Object

└ controlEnergiaEP.controller.JSFUtil

public class **JSFUtil**
 extends java.lang.Object

Constructor Summary

JSFUtil()

Method Summary

static void	<u>crearMensaje</u> (javax.faces.application.FacesMessage.Severity severidad, java.lang.String mensaje) Crea un mensaje JSF
static void	<u>crearMensajeERROR</u> (java.lang.String mensaje)
static void	<u>crearMensajeINFO</u> (java.lang.String mensaje)
static void	<u>crearMensajeWARN</u> (java.lang.String mensaje)

Aplicación de Escritorio TIMER

Package controlEnergiaEP.model.manager

Class ManagerControlEnergia

java.lang.Object

└ controlEnergiaEP.model.manager.ManagerControlEnergia

```
public class ManagerControlEnergia  
extends java.lang.Object
```

Clase que genera los CRUDS de las clases(Actividad, Registro, Dispositivo).

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

ManagerControlEnergia

```
public ManagerControlEnergia()
```

Method Detail

findAllActividad

```
public java.util.List<Actividad> findAllActividad()
```

Método finder para consulta de una actividad. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de las actividades ingresadas.

findActividadesActivas

```
public java.util.List<Actividad> findActividadesActivas()
```

Método finder para consulta de una actividad en la cual el estado es "true". Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de las actividades activas.

findActividadById

```
public Actividad findActividadById(java.lang.Integer idActividad) throws java.lang.Exception
```

Método finder para consulta de la actividad. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idActividad - código de la actividad que se desea buscar.

Returns:

la actividad encontrada.

Throws:

java.lang.Exception

actualizarActividad

public void actualizarActividad(Actividad actividad) throws java.lang.Exception

Actualiza la información de una actividad en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

actividad - Los datos de la actividad que se desea actualizar.

Throws:

java.lang.Exception

findAllRegistro

public java.util.List<Registro> findAllRegistro()

Método finder para consulta de Registro. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de registros.

findRegistroById

public Registro findRegistroById(java.lang.Integer idRegistro) throws java.lang.Exception

Método finder para consulta del registro. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idRegistro - código del registro que se desea buscar.

Returns:

el registro encontrado.

Throws:

java.lang.Exception

insertarRegistro

public void insertarRegistro(Registro re) throws java.lang.Exception

Guarda un nuevo registro en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

re - El nuevo registro.

Throws:

java.lang.Exception

actualizarRegistro

public void actualizarRegistro(Registro registro) throws java.lang.Exception

Actualiza la información de un registro en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

registro - Los datos del registro que se desea actualizar.

Throws:

java.lang.Exception

findSumaHoras

public double findSumaHoras(java.lang.Class clase, java.lang.Integer idregistro) throws java.lang.Exception

Busca en los Registros el id del registro ingresado como parámetro para obtener la suma de horas del mismo mediante las funciones extract(resta dos horas) y epoch(devuelve en milisegundos el resultado).

Parameters:

clase - clase en la que se va a realizar la búsqueda
idregistro - id del registro que se desea buscar.

Returns:

suma de horas.

Throws:

java.lang.Exception

findCountActividadesFalse

public int findCountActividadesFalse(java.lang.Class clase, Dispositivo iddispositivo) throws java.lang.Exception

Cuenta las actividades que se encuentran en estado falso; agrupadas por el id del dispositivo.

Parameters:

clase - clase en la que se va a realizar la búsqueda
iddispositivo - id del dispositivo a buscar.

Returns:

número de actividades que se encuentran en estado "false".

Throws:

java.lang.Exception

findCountActividades

public int findCountActividades(java.lang.Class clase, Dispositivo iddispositivo) throws java.lang.Exception

Cuenta todas las actividades que se han insertado; agrupadas por el id del dispositivo.

Parameters:

clase - clase en la que se va a realizar la búsqueda
iddispositivo - id del dispositivo a buscar.

Returns:

número de actividades.

Throws:

java.lang.Exception

Package controlEnergiaEP.model.controller

Class BeanTimer

java.lang.Object

└ controlEnergiaEP.model.controller.BeanTimer

```
public class BeanTimer  
extends java.lang.Object
```

Clase ejecutada por el timer que verifica a cada minuto si existe una actividad a ejecutar, ya sea para encender el dispositivo o para apagarlo.

Author:

Jenny Rea - YACHAY EP.

Constructor Detail

```
BeanTimer  
public BeanTimer()
```

Method Detail

estadoAP

```
public java.lang.String estadoAP() throws java.lang.Exception
```

Método que consulta todos los registros de las actividades activas; verificando: la fecha, el día y la hora en la que se creo cada una para así encender o apagar un dispositivo.

Returns:

cadena con mensajes sobre las acciones realizadas.

Throws:

java.lang.Exception

accionInsertarRegistro

```
public java.lang.String accionInsertarRegistro(java.lang.Integer idactividad, java.lang.String nombre,  
int puerto, java.lang.String ip)
```

Guarda un nuevo registro con la hora en la que se prende el dispositivo según la actividad, el nombre del dispositivo que enciende y envía la señal de apagado al equipo Raspberry.

Parameters:

idactividad - código de la actividad que se desea buscar.

nombre - del dispositivo en uso.

puerto - pin al que se encuentra conectado el dispositivo al equipo Raspberry.

ip - ip del equipo Raspberry.

Returns:

cadena con los mensajes de las acciones realizadas.

accionActualizarRegistro

```
public java.lang.String accionActualizarRegistro(java.lang.Integer idactividad, int puerto,  
java.lang.String ip)
```

Método que actualiza el registro respectivo con la hora en la que se apaga el dispositivo según la actividad, suma las horas de encendido/apagado y envía la señal de apagado al equipo Raspberry.

Parameters:

idactividad - código de la actividad que se desea buscar.

puerto - pin al que se encuentra conectado el dispositivo al equipo Raspberry.
ip - ip del equipo Raspberry.

Returns:

cadena con los mensajes de las acciones realizadas.

actionActualizarActividad

public void actionActualizarActividad(java.lang.Integer idactividad)

Método que actualiza el estado en "false" de la actividad cuando llega a su fin; en caso de que hayan finalizado todas las actividades de un dispositivo el estado del mismo se actualiza a "false".

Parameters:

idactividad - código de la actividad que se desea actualizar.

Package vista**Class VistaTimer**

java.lang.Object

└ vista

java.lang.Object

java.awt.Component

java.awt.Container

java.awt.Window

java.awt.Frame

javax.swing.JFrame

vista.VistaTimer

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,

javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

```
public class VistaTimer
extends javax.swing.JFrame
```

Vista del timer en la cual aparece las acciones que ejecutan en el Sistema mediante un JTextArea.

Author:

Jenny Rea - YACHAY EP.

See Also:

[Serialized Form](#)

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame javax.swing.JFrame.AccessibleJFrame

Nested classes/interfaces inherited from class java.awt.Frame java.awt.Frame.AccessibleAWTFrame

Field Summary

Fields	
Modifier and Type	Field and Description
static javax.swing.JTextArea	<u>jTextArea1</u>

Constructor Summary

Constructors
Constructor and Description
<u>VistaTimer()</u> Método que permite crear un nuevo timer y crear la tarea en 1000 ms de inicio y cada minuto se vuelve a ejecutar la misma.

Method Summary

Methods	
Modifier and Type	Method and Description
java.lang.String	<u>getEstado()</u>
static void	<u>main</u> (java.lang.String[] args)
void	<u>setEstado</u> (java.lang.String estado)

Method Detail

main public static void main(java.lang.String[] args) Parameters: args - the command line arguments

DICCIONARIO DE DATOS

Generated: 09/07/2015 23:20:00

Server: PostgreSQL 9.3 (localhost:5432)

Database: controlEnergia

Schema: public

Tabla parámetro

Columnas

Name	Data type	Not Null?	Primar y key?	Default	Comment
nombre	character varying(2)	Yes	Yes		Clave primaria de la tabla, de longitud 2. En este campo se colocarán las iniciales del parámetro.
valor	character varying(50)	Yes	No		Este campo contiene el valor del parámetro a ingresar.
descripcion	character varying(100)	Yes	No		Aquí se agrega una pequeña descripción para mencionar el uso del parámetro.
clavepordefecto	character varying(150)	Yes	NO		Campo que posee la clave por defecto para restablecer la contraseña del funcionario.

Constraints

Name	Type	Definition
parametro_pkey	Primary key	(nombre)

Tabla tipousuario

Columnas

Name	Data type	Not Null?	Primary key?	Default	Comment
idtipousuario	integer	Yes	Yes		Generada automáticamente por la secuencia seqtipousuario y controlada en la aplicación (este

					campo es clave primaria de la tabla).
usuario	character varying(20)	Yes	No		Tipo de usuario que existe en el sistema de control de energía eléctrica.

Constraints

Name	Type	Definition
tipousuario_pkey	Primary key	(idtipousuario)

Tabla funcionario

Columnas

Name	Data type	Not Null?	Primary key?	Default	Comment
cedula	character varying(10)	Yes	Yes		
nombres	character varying(50)	Yes	No		
apellidos	character varying(50)	Yes	No		
correo	character varying(50)	Yes	No		Campo que contiene el correo electrónico.
clave	character varying(15)	Yes	No		Este campo es para validar el ingreso al sistema en el cual por defecto será el valor que el Administrador coloque en el campo de la tabla parámetros (PW).
idtipousuario	integer	Yes	No		
estado	boolean	Yes	No	true	Campo de tipo boolean que cuando está en true significa que el funcionario está habilitado para ingresar al sistema y false en el caso contrario (por defecto todos los funcionarios se ingresan con true).

Constraints

Name	Type	Definition
funcionario_pkey	Primary key	(cedula)
funcionario_idtipousuario_fkey	Foreign key	(idtipousuario) REFERENCES tipousuario (idtipousuario) MATCH SIMPLE ON UPDATE NO ACTION ON

DELETE NO ACTION

Tabla evento

Columnas

Name	Data type	Not Null?	Primary key?	Default	Comment
codigoevento	integer	Yes	Yes		Clave primaria de la tabla en la cual se genera automáticamente con la secuencia creada llamada: seevento (Se controla el aumento en la aplicación en la entitie).
fecha_evento	date	Yes	No		La fecha actual en la que se efectúa la acción.
acción	character varying(100)	Yes	No		Actividad que realiza el funcionario en el sistema (insertar, actualizar o eliminar).
ipcomputador	character varying(50)	No	No		Campo que almacena el Usuario usado en el SO y la ip del computador.
detalle	text	Yes	No		
cedula	character varying(10)	Yes	No		Campo para guardar la cédula de identidad del funcionario que realiza la acción, siendo su longitud de 10 caracteres.

Constraints

Name	Type	Definition
evento_pkey	Primary key	(codigoevento)
evento_cedula_fkey	Foreign key	(cedula) REFERENCES functionaries (cedula) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

Tabla modelo

Columnas

Name	Data type	Not Null?	Primary key?	Default	Comment
idmodelo	integer	Yes	Yes		Clave primaria de la tabla en la cual se genera automáticamente mediante la secuencia (seqmodelo) controlada en la

					aplicación.
nombre	character varying(50)	Yes	No		Campo para ingresar el nombre del modelo del Raspberry Pi.
descripcion	character varying(100)	No	No		
pin1	integer	Yes	No		Campo obligatorio para ingresar el pin del GPIO del Raspberry Pi al que se va a conectar el dispositivo. Nota: los otros campos con prefijo pin también cumplen la misma función.
pin2	integer	Yes	No		
pin3	integer	No	No		
pin4	integer	No	No		

Constraints

Name	Type	Definition	Comment
modelo_pkey	Primary key	(idmodelo)	

Tabla raspberry

Columnas

Name	Data type	Not Null?	Primary key?	Default	Comment
idraspberry	integer	Yes	Yes		Clave primaria de la tabla en la cual se genera automáticamente mediante la secuencia (seqraspberry) controlada en la aplicación.
nombre	character varying(100)	Yes	No		
ipraspberry	character varying(50)	Yes	No		Campo para almacenar la ip asignada al Raspberry Pi
ubicación	character varying(100)	Yes	No		Este campo sirve para ingresar en que sitio se encuentra ubicado el Raspberry Pi.
descripcion	text	Yes	No		
observación	text	No	No		
idmodelo	integer	Yes	No		

Constraints

Name	Type	Definition
------	------	------------

raspberry_pkey	Primary key	(idraspberry)
idmodelo	Foreign key	(idmodelo) REFERENCES modelo (idmodelo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

Tabla tipodispositivo

Columnas

Name	Data type	Not Null ?	Primary key?	Default	Comment
idtipodispositivo	integer	Yes	Yes		Clave primaria del tipo de dispositivo que se genera automáticamente por la secuencia (seqtipodispositivo)
nombre	character varying(50)	No	No		
consumo	integer	No	No		Número que determina cual es el consumo del dispositivo en (KWTTTS).

Constraints

Name	Type	Definition
tipodispositivo_pkey	Primary key	(idtipodispositivo)

Tabla dispositivo

Columnas

Name	Data type	Not Null?	Primary key?	Default	Comment
iddispositivo	integer	Yes	Yes		Clave primaria de la tabla en la cual se genera automáticamente mediante la secuencia (seqdispositivo) controlada en la aplicación.
idtipodispositivo	integer	Yes	No		
ubicación	character varying(100)	Yes	No		En este campo se ingresa el sitio referencial de donde se encuentra el dispositivo.
descripcion	text	Yes	No		
observación	text	No	No		
idraspberry	integer	Yes	No		

estado	boolean	No	No	false	Campo de tipo boolean que muestra si el dispositivo esta prendido (true) o apagado (false).
puerto	integer	No	No		En este campo se ingresa el pin por el cual va a estar conectado dicho dispositivo al GPIO del Raspberry Pi.

Constraints

Name	Type	Definition
dispositivo_pkey	Primary key	(iddispositivo)
dispositivo_idraspberry_fkey	Foreign key	(idraspberry) REFERENCES raspberr (idraspberry) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION
dispositivo_idtipodispositivo	Foreign key	(idtipodispositivo) REFERENCES tipodispositivo (idtipodispositivo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

Tabla actividad

Columns

Name	Data type	Not Null?	Primary key?	Default	Comment
idactividad	integer	Yes	Yes		Clave primaria de la tabla en la cual se genera automáticamente mediante la secuencia (seqactividad) controlada en la aplicación.
fechaActual	date	Yes	No		Fecha Actual en la que se realiza la actividad.
hora_desde	time without time zone	No	No		Hora en la que se inicia la actividad; es decir la hora que se va a encender el dispositivo elegido por el usuario.
hora_hasta	time without time zone	No	No		Hora de finalización de la actividad; es decir la hora en la que se apaga el dispositivo (deja de pasar corriente eléctrica).
fechaInicio	date	Yes	No		Fecha en la que inicia la actividad.
fechaFin	date	No	No		Fecha en la que finaliza la actividad.
iddispositivo	integer	Yes	No		

lunes	boolean	No	No		Campo de tipo boolean que representa al día Lunes que si está seleccionado está en true, caso contrario está en false. Este campo sirve para seleccionar que en este día se va a efectuar la actividad. Nota: los demás campos con los nombres de los días de la semana cumplen la misma función.
martes	boolean	No	No		
miércoles	boolean	No	No		
jueves	boolean	No	No		
viernes	boolean	No	No		
sábado	boolean	No	No		
domingo	boolean	No	No		
observación	text	No	No		
nombre	character varying(50)	No	No		

Constraints

Name	Type	Definition
actividad_pkey	Primary key	(idactividad)
iddispositivo	Foreign key	(iddispositivo) REFERENCES dispositivo (iddispositivo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

Tabla registro

Columns

Name	Data type	Not Null?	Primary key?	Comment
idregistro	integer	Yes	Yes	Clave primaria de la tabla que se genera automáticamente con la secuencia (seqregistro).
idactividad	integer	No	No	
nombre	character varying(50)	No	No	Campo que contiene el nombre del dispositivo de la actividad vinculada.
inicio	timestamp without time zone	No	No	Este campo registra la fecha actual y la hora en la que debe prenderse el dispositivo según la actividad creada.
fin	timestamp without time zone	No	No	Campo que registra la fecha actual y la hora en la que debe apagarse el dispositivo según la actividad creada.
totalH	numeric(12,2)	No	No	Atributo que guarda la suma de horas del campo "inicio" y "fin".

Constraints

Name	Type	Definition
registro_pkey	Primary key	(idregistro)
registro_idactividad_fkey	Foreign key	(idactividad) REFERENCES actividad (idactividad) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION