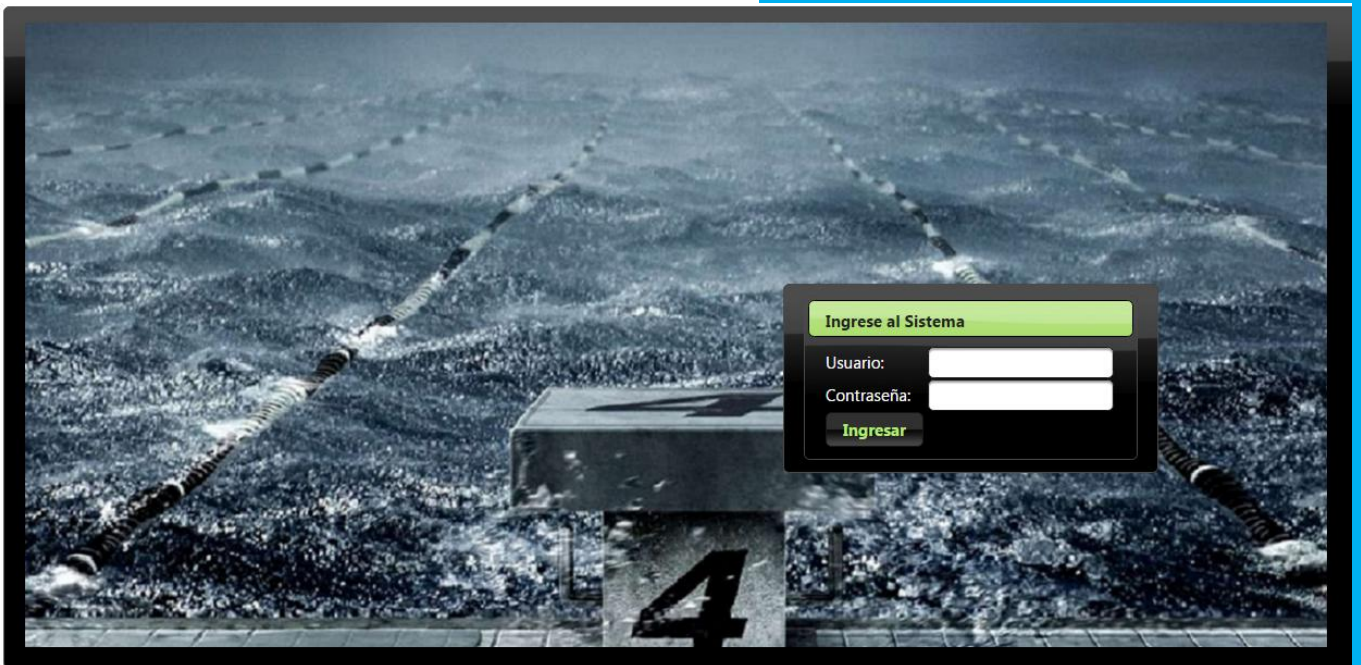


SISTEMA PARA LA
PLANIFICACIÓN DE
ENTRENAMIENTOS Y GESTIÓN
DE COMPETENCIAS EN EL ÁREA
DE LA NATACIÓN

2016

MANUAL TÉCNICO



MAURICIO CARRILLO
UNIVERSIDAD TÉCNICA
DEL NORTE

6-4-2016

ÍNDICE

DESCRIPCIÓN DEL SISTEMA	4
HERRAMIENTAS APLICADAS.....	4
JDK 7.0	4
POSTGRES 9.3.....	4
APACHE TOMCAT 7.055.....	4
ECLIPSE.....	4
OPENSIFT.....	4
AXURE.....	5
GOOGLE CHROME.....	5
FRAMEWORKS	5
PRIMEFACES	5
JPA ENTITIES	5
DISEÑO DE BASE DE DATOS	6
DICCIONARIO DE DATOS.....	7
TABLA CAMPEONATO	7
TABLA CATEGORÍA.....	7
TABLA CLUBES.....	7
TABLA DEPORTISTAS	8
TABLA ENTRENADOR.....	8
TABLA ENTRENAMIENTODIARIO	9
TABLA ENTRENO.....	9
TABLA ESTILOS	10
TABLA GÉNERO	10
TABLA HORARIO	10
TABLA INSCRIPCIÓN.....	11
TABLA LOGIN.....	11
TABLA NADADORES.....	11
TABLA NIVELES.....	12
TABLA POSICIONES	12
TABLA PRUEBAS	13
TABLA SERIES.....	13

TABLA TEMPORADAS	13
TABLA TIPOENTRENAMIENTO	14
TABLA TOMATIEMPOS	14
DESCRIPCIÓN DE CLASES Y MÉTODOS.....	15
PACKAGES COMPETENCIA.MODELO.MANAGER	15
CLASS CAMPEONATO	15
CLASS CATEGORIA	16
CLASS CLUB	17
CLASS DEPORTISA	18
CLASS ENTRENADOR	20
CLASS ENTRENAMIENTODIARIO	22
CLASS ENTRENO	23
CLASS ESTILO	24
CLASS GENERO.....	25
CLASS HORARIO.....	26
CLASS INSCRIPCION	27
CLASS LOGGIN.....	29
CLASS NADADORE.....	30
CLASS NIVELE.....	31
CLASS POSICIONE	32
CLASS PRUEBA	33
CLASS SERY	35
CLASS TEMPORADA	36
CLASS TIPOENTRENAMIENTO	37
CLASS TOMATIEMPO	38
PACKAGES COMPETENCIA.MODELO.MANAGER	39
CLASS MANAGERCOMPETENCIA	40
CLASS MANAGERDAO	51
CLASS MANAGERENTRENAMIENTO.....	55
PACKAGES COMPETENCIA.CONTROLLER	63
CLASS BEANAVANCES	63
CLASS BEANCAMPEONATO	65
CLASS BEANCLUB.....	66

CLASS BEANDEPORTISTA.....	67
CLASS BEANENTRENADOR.....	69
CLASS BEANENTRENAMIENTO.....	70
CLASS BEANINSCRIPCIONE.....	72
CLASS BEANLOGIN.....	73
CLASS BEANNADADORE.....	74
CLASS BEANPOSICIONES.....	76
CLASS BEANREPORTE.....	76
CLASS BEANSERIES.....	77
CLASS BEANTEMPORADA.....	78
ALOJAMIENTO DEL SISTEMA EN OPENSIFT.....	80

DESCRIPCIÓN DEL SISTEMA

Tras las distintas necesidades obtenidas en el club de natación UTN sobre la falta de un sistema que permite el control y administración de entrenamientos y competencias se crea el presente sistema: “SISTEMA DE PLANIFICACIÓN DE ENTRENAMIENTOS”, con el fin de agilizar los procesos y llevar un control organizado de los entrenamientos, registro de: entrenadores, deportistas, tiempos en distintas pruebas y avance además de la generación y obtención de resultados en el proceso de competencias. Adicional a lo mencionado el sistema se encuentra alojado en una plataforma Paas (Openshift) lo que permite su manipulación desde cualquier equipo con acceso a internet.

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 9.3

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 Mars 1.0. Herramienta de programación de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido".

OPENSIFT

Para el alojamiento del sistema se utiliza una plataforma de servicio (PaaS) que es Openshift, el mismo que es gratuito.

AXURE

Para la creación de prototipos del sistema se utilizó la herramienta Axure la misma que cuenta con variedad de opciones lo cual facilita su diseño.

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).

FRAMEWORKS

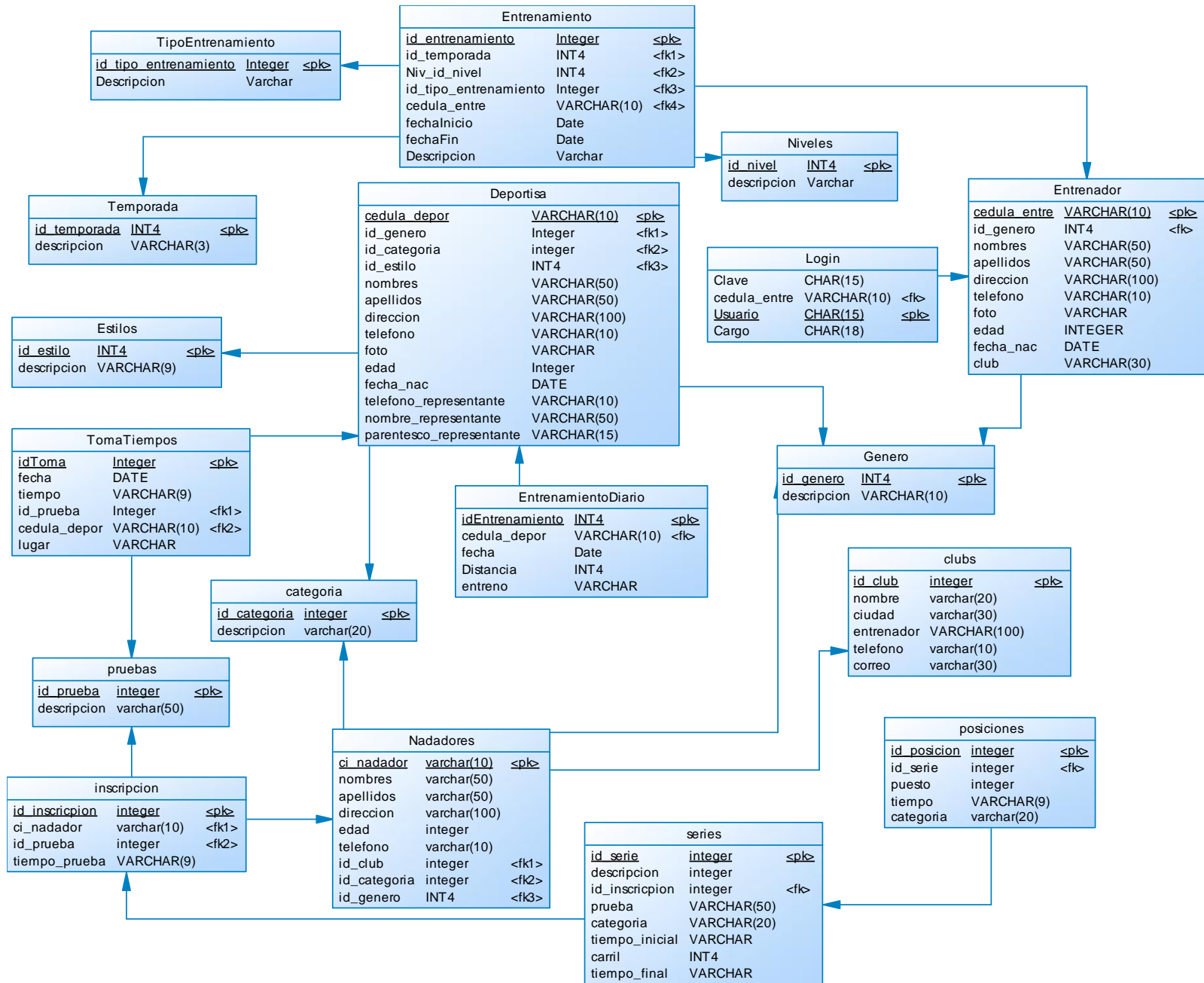
PRIMEFACES

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).

DISEÑO DE BASE DE DATOS



DICCIONARIO DE DATOS

TABLA CAMPEONATO

Nombre	Tipo	Nulo	Clave Primaria
cod_campeonato	integer	Si	Si
nombre	character varying	No	No
fecha_inicio	date	No	No
fecha_fin	date	No	No
estado	character varying	No	No

Claves

Nombre	Tipo	Definición
campeonato_pkey	Clave Primaria	(cod_campeonato)

TABLA CATEGORÍA

Nombre	Tipo	Nulo	Clave Primaria
id_categoria	integer	Si	Si
descripcion	character varying(20)	No	No

Claves

Nombre	Tipo	Definición
pk_categoria	Clave Primaria	(id_categoria)

TABLA CLUBES

Nombre	Tipo	Nulo	Clave Primaria
id_club	integer	Si	Si
nombre	character varying(20)	No	No
ciudad	character varying(30)	No	No
entrenador	character varying(100)	No	No
telefono	character varying(10)	No	No
correo	character varying(30)	No	No

Claves

Nombre	Tipo	Definición
pk_clubs	Clave Primaria	(id_club)

TABLA DEPORTISTAS

Nombre	Tipo	Nulo	Clave Primaria
cedula_depor	character varying(10)	Si	Si
id_genero	integer	No	No
id_categoria	integer	No	No
id_estilo	integer	No	No
nombres	character varying(50)	No	No
apellidos	character varying(50)	No	No
direccion	character varying(100)	No	No
telefono	character varying(10)	No	No
foto	character varying	No	No
edad	integer	No	No
fecha_nac	date	No	No
telefono_representante	character varying(10)	No	No
nombre_representante	character varying(50)	No	No
parentesco_representante	character varying(15)	No	No

Claves

Nombre	Tipo	Definición
pk_deportisa	Clave Primaria	(cedula_depor)
fk_deportis_referencia_categoria	Clave Foránea	(id_categoria) REFERENCES categoria (id_categoria) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_deportis_referencia_estilos	Clave Foránea	(id_estilo) REFERENCES estilos (id_estilo) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_deportis_referencia_genero	Clave Foránea	(id_genero) REFERENCES genero (id_genero) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA ENTRENADOR

Nombre	Tipo	Nulo	Clave Primaria
cedula_entre	character varying(10)	Si	Si
id_genero	integer	No	No
nombres	character varying(50)	No	No
apellidos	character varying(50)	No	No
direccion	character varying(100)	No	No
telefono	character varying(10)	No	No
foto	character varying	No	No
edad	integer	No	No
fecha_nac	date	No	No
club	character varying(30)	No	No

Claves

Nombre	Tipo	Definición
pk_entrenador	Clave Primaria	(cedula_entre)
fk_entrenad_re ference_genero	Clave Foránea	(id_genero) REFERENCES genero (id_genero) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA ENTRENAMIENTODIARIO

Nombre	Tipo	Nulo	Clave Primaria
identrenamiento	integer	Si	Si
cedula_depor fecha	character varying(10) date	No	No
idcond	integer	No	No

Claves

Nombre	Tipo	Definición
pk_entrenamie ntodiaro	Clave Primaria	(identrenamiento)
fk_entrenam_r eference_depor tis	Clave Foránea	(cedula_depor) REFERENCES deportisa (cedula_depor) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_entrenam_r eference_entre no	Clave Foránea	(idcond) REFERENCES entreno (idcond) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

TABLA ENTRENO

Nombre	Tipo	Nulo	Clave Primaria
idcond	integer	Si	Si
descripcion	character varying	No	No
fechainicio	date	No	No
fechafin	date	No	No
id_nivel	integer	No	No
id_tipo_entrenamiento	integer	No	No
distancia	integer	No	No
usuario	character varying(15)	No	No
id_horario	integer	No	No
id_temporada	integer	No	No

Claves

Nombre	Tipo	Definición
entreno_pkey	Clave Primaria	(idcond)
fk_entrenam_ reference_hor ario	Clave Foránea	(id_horario) REFERENCES horario (id_horario) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

fk_entrenam_ reference_log gin	Clave Foránea	(usuario) REFERENCES loggin (usuario) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION
fk_entrenam_ reference_niv eles	Clave Foránea	(id_nivel) REFERENCES niveles (id_nivel) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_entrenam_ reference_te mporada	Clave Foránea	(id_temporada) REFERENCES temporada (id_temporada) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION
fk_entrenam_ reference_tip oentr	Clave Foránea	(id_tipo_entrenamiento) REFERENCES tipoentrenamiento (id_tipo_entrenamiento) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA ESTILOS

Nombre	Tipo	Nulo	Clave Primaria
id_estilo	integer	Si	Si
descripcion	character varying(9)	No	No

Claves

Nombre	Tipo	Definición
pk_estilos	Clave Primaria	(id_estilo)

TABLA GÉNERO

Nombre	Tipo	Nulo	Clave Primaria
id_genero	integer	Si	Si
descripcion	character varying(10)	No	No

Claves

Nombre	Tipo	Definición
pk_genero	Clave Primaria	(id_genero)

TABLA HORARIO

Nombre	Tipo	Nulo	Clave Primaria
id_horario	integer	Si	Si
tipo	character varying	No	No

Claves

Nombre	Tipo	Definición
horario_pkey	Clave Primaria	(id_horario)

TABLA INSCRIPCIÓN

Nombre	Tipo	Nulo	Clave Primaria
id_inscripcion	integer	Si	Si
ci_nadador	character varying(10)	No	No
id_prueba	integer	No	No
tiempo_prueba	character varying(9)	No	No
cod_campeonato	integer	No	No

Claves

Nombre	Tipo	Definición
pk_inscripcion	Clave Primaria	(id_inscripcion)
fk_inscripc_ref erence_campe onato	Clave Foránea	(cod_campeonato) REFERENCES campeonato (cod_campeonato) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION
fk_inscripc_ref erence_nadado re	Clave Foránea	(ci_nadador) REFERENCES nadadores (ci_nadador) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_inscripc_ref erence_prueba s	Clave Foránea	(id_prueba) REFERENCES pruebas (id_prueba) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA LOGIN

Nombre	Tipo	Nulo	Clave Primaria
clave	character varying	Si	No
cedula_entre usuario	character varying(10)	No	No
cargo	character(15)	Si	Si
	character(18)	No	No

Claves

Nombre	Tipo	Definición
login_pkey	Clave Primaria	(usuario)
fk_login_refe rence_entrenad	Clave Foránea	(cedula_entre) REFERENCES entrenador (cedula_entre) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA NADADORES

Nombre	Tipo	Nulo	Clave Primaria
ci_nadador	character varying(10)	Si	Si
nombres	character varying(50)	No	No
apellidos	character varying(50)	No	No
direccion	character varying(100)	No	No
edad	integer	No	No

telefono	character varying(10)	No	No
id_club	integer	No	No
id_categoria	integer	No	No
id_genero	integer	No	No

Claves

Nombre	Tipo	Definición
pk_nadadores	Clave Primaria	(ci_nadador)
fk_nadadore_referencia_categoria	Clave Foránea	(id_categoria) REFERENCES categoria (id_categoria) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_nadadore_referencia_clubs	Clave Foránea	(id_club) REFERENCES clubs (id_club) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_nadadore_referencia_genero	Clave Foránea	(id_genero) REFERENCES genero (id_genero) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA NIVELES

Nombre	Tipo	Nulo	Clave Primaria
id_nivel	integer	Si	Si
descripcion	character varying(3)	No	No

Claves

Nombre	Tipo	Definición
pk_niveles	Clave Primaria	(id_nivel)

TABLA POSICIONES

Nombre	Tipo	Nulo	Clave Primaria
id_posicion	integer	Si	Si
id_serie	integer	No	No
puesto	integer	No	No
tiempo	character varying(9)	No	No
categoria	character varying(20)	No	No
cod_campeonato	integer	No	No

Claves

Nombre	Tipo	Definición
pk_posiciones	Clave Primaria	(id_posicion)
fk_posicion_referencia_campeonato	Clave Foránea	(cod_campeonato) REFERENCES campeonato (cod_campeonato) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION
fk_posicion_referencia_series	Clave Foránea	(id_serie) REFERENCES series (id_serie) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA PRUEBAS

Nombre	Tipo	Nulo	Clave Primaria
id_prueba	integer	Si	Si
descripcion	character varying(50)	No	No

Claves

Nombre	Tipo	Definición
pk_pruebas	Clave Primaria	(id_prueba)

TABLA SERIES

Nombre	Tipo	Nulo	Clave Primaria
id_serie	integer	Si	Si
descripcion	integer	No	No
id_inscripcion	integer	No	No
prueba	character varying(50)	No	No
categoria	character varying(20)	No	No
tiempo_inicial	character varying	No	No
carril	integer	No	No
tiempo_final	character varying	No	No
cod_campeonato	integer	No	No

Claves

Nombre	Tipo	Definición
pk_series	Clave Primaria	(id_serie)
fk_series_referencia_campeonato	Clave Foránea	(cod_campeonato) REFERENCES campeonato (cod_campeonato) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION
fk_series_referencia_inscripcion	Clave Foránea	(id_inscripcion) REFERENCES inscripcion (id_inscripcion) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

TABLA TEMPORADAS

Nombre	Tipo	Nulo	Clave Primaria
id_temporada	integer	Si	Si
descripcion	character varying	No	No
fecha_inicio	date	No	No
fecha_fin	date	No	No
objetivo	character varying	No	No

Claves

Nombre	Tipo	Definición
temporada_pkey	Clave Primaria	(id_temporada)

TABLA TIPOENTRENAMIENTO

Nombre	Tipo	Nulo	Clave Primaria
id_tipo_entrenamiento	integer	Si	Si
descripcion	character varying(20)	No	No

Claves

Nombre	Tipo	Definición
pk_tipoentrenamiento	Clave Primaria	(id_tipo_entrenamiento)

TABLA TOMATIEMPOS

Nombre	Tipo	Nulo	Clave Primaria
idtoma	integer	Si	Si
fecha	date	No	No
tiempo	character varying(9)	No	No
id_prueba	integer	No	No
cedula_depor	character varying(10)	No	No
lugar	character varying	No	No

Claves

Nombre	Tipo	Definición
pk_tomatiempos	Clave Primaria	(idtoma)
fk_tomatiem_reference_deportis	Clave Foránea	(cedula_depor) REFERENCES deportisa (cedula_depor) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT
fk_tomatiem_reference_pruebas	Clave Foránea	(id_prueba) REFERENCES pruebas (id_prueba) MATCH SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT

DESCRIPCIÓN DE CLASES Y MÉTODOS

PACKAGES COMPETENCIA.MODELO.MANAGER

En este paquete se encuentran las distintas clases y métodos que permiten la interacción con la base de datos.

CLASS CAMPEONATO

java.lang.Object

competencia.modelo.Campeonato

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Campeonato
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
Campeonato()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Inscripcion	addInscripcion(Inscripcion inscripcion)
Posicione	addPosicione(Posicione posicion)
Sery	addSery(Sery sery)
java.lang.Integer	getCodCampeonato()
java.lang.String	getEstado()
java.util.Date	getFechaInicio()
java.util.Date	getFechFin()
java.util.List< Inscripcion >	getInscripciones()
java.lang.String	getNombre()
java.util.List< Posicione >	getPosiciones()

java.util.List< Sery >	getSeries()
Inscripcion	removeInscripcion(Inscripcion inscripcion)
Posicione	removePosicione(Posicione posicione)
Sery	removeSery(Sery sery)
void	setCodCampeonato(java.lang.Integer codCampeonato)
void	setEstado(java.lang.String estado)
void	setFechaInicio(java.util.Date fechaInicio)
void	setFechFin(java.util.Date fechFin)
void	setInscripciones(java.util.List<Inscripcion> inscripciones)
void	setNombre(java.lang.String nombre)
void	setPosiciones(java.util.List<Posicione> posiciones)
void	setSeries(java.util.List<Sery> series)

Methods inherited from class java.lang.Object

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

CLASS CATEGORIA

java.lang.Object
competencia.modelo.Categoria

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Categoria
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
Categoria()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
	addDeportisa(Deportisa deportisa)
	addNadadore(Nadadore nadadore)
java.util.List< Deportisa >	getDeportisas()
java.lang.String	getDescripcion()
java.lang.Integer	getIdCategoria()
java.util.List< Nadadore >	getNadadores()

Deportisa	removeDeportisa (Deportisa deportisa)
Nadadore	removeNadadore (Nadadore nadadore)
void	setDeportisas (java.util.List< Deportisa > deportisas)
void	setDescripcion (java.lang.String descripcion)
void	setIdCategoria (java.lang.Integer idCategoria)
void	setNadadores (java.util.List< Nadadore > nadadores)

Methods inherited from class java.lang.Object

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

CLASS CLUB

java.lang.Object
 competencia.modelo.Club

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Club
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description

Club()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
--------------------	----------------------

Nadadore	addNadadore (Nadadore nadadore)
java.lang.String	getCiudad ()
java.lang.String	getCorreo ()

java.lang.String	getEntrenador()
java.lang.Integer	getIdClub()
java.util.List< Nadadore >	getNadadores()
java.lang.String	getNombre()
java.lang.String	getTelefono()
Nadadore	removeNadadore(Nadadore nadadore)
void	setCiudad (java.lang.String ciudad)
void	setCorreo (java.lang.String correo)
void	setEntrenador (java.lang.String entrenador)
void	setIdClub (java.lang.Integer idClub)
void	setNadadores (java.util.List< Nadadore > nadadores)
void	setNombre (java.lang.String nombre)
void	setTelefono (java.lang.String telefono)

Methods inherited from class java.lang.Object
--

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

CLASS DEPORTISA

```
java.lang.Object
  competencia.modelo.Deportisa
```

All Implemented Interfaces:

```
java.io.Serializable
```

```
@Entity
public class Deportisa
  extends java.lang.Object
  implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
Deportisa()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entrenamientodiario	addEntrenamientodiario(Entrenamientodiario entrenamientodiario)
Tomatiempo	addTomatiempo(Tomatiempo tomatiempo)
java.lang.String	getApellidos()
Categoria	getCategoria()
java.lang.String	getCédulaDepor()
java.lang.String	getDireccion()
java.lang.Integer	getEdad()
java.util.List< Entrenamientodiario >	getEntrenamientodiaros()
Estilo	getEstilo()
java.util.Date	getFechaNac()
java.lang.String	getFoto()
Genero	getGenero()
java.lang.String	getNombreRepresentante()
java.lang.String	getNombres()
java.lang.String	getParentescoRepresentante()
java.lang.String	getTelefono()
java.lang.String	getTelefonoRepresentante()
java.util.List< Tomatiempo >	getTomatiempos()
Entrenamientodiario	removeEntrenamientodiario(Entrenamientodiario entrenamientodiario)
Tomatiempo	removeTomatiempo(Tomatiempo tomatiempo)

```

void setApellidos(java.lang.String apellidos)
void setCategoria(Categoria categoria)
void setCédulaDepor(java.lang.String cédulaDepor)
void setDireccion(java.lang.String direccion)
void setEdad(java.lang.Integer edad)
void setEntrenamientodiarios(java.util.List<Entrenamientodiario>
entrenamientodiarios)
void setEstilo(Estilo estilo)
void setFechaNac(java.util.Date fechaNac)
void setFoto(java.lang.String foto)
void setGenero(Genero genero)
void setNombreRepresentante(java.lang.String nombreRepresentante
)
void setNombres(java.lang.String nombres)
void setParentescoRepresentante(java.lang.String parentescoReprese
ntante)
void setTelefono(java.lang.String telefono)
void setTelefonoRepresentante(java.lang.String telefonoRepresentant
e)
void setTomatiempos(java.util.List<Tomatiempo> tomatiempos)

```

Methods inherited from class java.lang.Object
--

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

CLASS ENTRENADOR

```

java.lang.Object
  competencia.modelo.Entrenador

```

All Implemented Interfaces:

```

  java.io.Serializable

```

```

@Entity
public class Entrenador
  extends java.lang.Object

```

implements java.io.Serializable

RESUMEN DE CONSTRUCTORES

Constructor and Description

<u>Entrenador()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Loggin	addLoggin(Loggin loggin)
java.lang.String	getApellidos()
java.lang.String	getCédulaEntre()
java.lang.String	getClub()
java.lang.String	getDireccion()
java.lang.Integer	getEdad()
java.util.Date	getFechaNac()
java.lang.String	getFoto()
Genero	getGenero()
java.util.List<Loggin>	getLoggins()
java.lang.String	getNombres()
java.lang.String	getTelefono()
Loggin	removeLoggin(Loggin loggin)
void	setApellidos(java.lang.String apellidos)
void	setCédulaEntre(java.lang.String cédulaEntre)
void	setClub(java.lang.String club)
void	setDireccion(java.lang.String direccion)
void	setEdad(java.lang.Integer edad)
void	setFechaNac(java.util.Date fechaNac)

```

void | setFoto(java.lang.String foto)
void | setGenero(Genero genero)
void | setLoggins(java.util.List<Loggin> loggins)
void | setNombres(java.lang.String nombres)
void | setTelefono(java.lang.String telefono)

```

Methods inherited from class java.lang.Object

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

CLASS ENTRENAMIENTODIARIO

```

java.lang.Object
    competencia.modelo.Entrenamientodiario

```

All Implemented Interfaces:

```

    java.io.Serializable

```

```

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

```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Entrenamientodiario()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
--------------------	----------------------

Deportisa	getDeportisa()
Entreno	getEntreno()
java.util.Date	getFecha()
java.lang.Integer	getIdentrenamiento()
void	setDeportisa(Deportisa deportisa)

```

void setEntreno(Entreno entreno)
void setFecha(java.util.Date fecha)
void setIdentrenamiento(java.lang.Integer identrenamiento)

```

Methods inherited from class java.lang.Object

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

CLASS ENTRENO

java.lang.Object
competencia.modelo.Entreno

All Implemented Interfaces:

java.io.Serializable

```

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

```

RESUMEN DE CONSTRUCTORES

Constructor and Description

Entreno()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entrenamientodiario	addEntrenamientodiario(Entrenamientodiario entrenamientodiario)
java.lang.String	getDescripcion()
java.lang.Integer	getDistancia()
java.util.List< Entrenamientodiario >	getEntrenamientodiaros()
java.util.Date	getFechafin()

java.util.Date	getFechainicio()
Horario	getHorario()
java.lang.Integer	getIdcond()
Loggin	getLoggin()
Nivele	getNivele()
Temporada	getTemporada()
Tipoentrenamiento	getTipoentrenamiento()
Entrenamientodiario	removeEntrenamientodiario(Entrenamientodiario entrenamien todiario)
void	setDescription(java.lang.String descripcion)
void	setDistancia(java.lang.Integer distancia)
void	setEntrenamientodiarios(java.util.List<Entrenamientodiario> entrenamientodiarios)
void	setFechafin(java.util.Date fechafin)
void	setFechainicio(java.util.Date fechainicio)
void	setHorario(Horario horario)
void	setIdcond(java.lang.Integer idcond)
void	setLoggin(Loggin loggin)
void	setNivele(Nivele nivele)
void	setTemporada(Temporada temporada)
void	setTipoentrenamiento(Tipoentrenamiento tipoentrenamiento)

Methods inherited from class java.lang.Object

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

CLASS ESTILO

java.lang.Object
competencia.modelo.Estilo

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Estilo
extends java.lang.Object
```

implements java.io.Serializable

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Estilo()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Deportisa	addDeportisa(Deportisa deportisa)
java.util.List< Deportisa >	getDeportisas()
java.lang.String	getDescripcion()
java.lang.Integer	getIdEstilo()
Deportisa	removeDeportisa(Deportisa deportisa)
void	setDeportisas(java.util.List<Deportisa> deportisas)
void	setDescripcion(java.lang.String descripcion)
void	setIdEstilo(java.lang.Integer idEstilo)

Methods inherited from class java.lang.Object

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

CLASS GENERO

```
java.lang.Object
    competencia.modelo.Genero
```

All Implemented Interfaces:

```
java.io.Serializable
```

```
@Entity
public class Genero
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
Genero()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO MÉTODO Y DESCRIPCIÓN

Deportisa	addDeportisa(Deportisa deportisa)
Entrenador	addEntrenador(Entrenador entrenador)
Nadadore	addNadadore(Nadadore nadadore)
java.util.List< Deportisa >	getDeportisas()
java.lang.String	getDescripcion()
java.util.List< Entrenador >	getEntrenadores()
java.lang.Integer	getIdGenero()
java.util.List< Nadadore >	getNadadores()
Deportisa	removeDeportisa(Deportisa deportisa)
Entrenador	removeEntrenador(Entrenador entrenador)
Nadadore	removeNadadore(Nadadore nadadore)
void	setDeportisas(java.util.List<Deportisa> deportisas)
void	setDescripcion(java.lang.String descripcion)
void	setEntrenadores(java.util.List<Entrenador> entrenadores)
void	setIdGenero(java.lang.Integer idGenero)
void	setNadadores(java.util.List<Nadadore> nadadores)

Methods inherited from class java.lang.Object
--

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

CLASS HORARIO

java.lang.Object
competencia.modelo.Horario

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Horario
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Horario()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entreno	addEntreno(Entreno entreno)
java.util.List< Entreno >	getEntrenos()
java.lang.Integer	getIdHorario()
java.lang.String	getTipo()
Entreno	removeEntreno(Entreno entreno)
void	setEntrenos(java.util.List<Entreno> entrenos)
void	setIdHorario(java.lang.Integer idHorario)
void	setTipo(java.lang.String tipo)

Methods inherited from class java.lang.Object

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

CLASS INSCRIPCION

```
java.lang.Object
competencia.modelo.Inscripcion
```

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Inscripcion
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Inscripcion()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Sery	addSery(Sery sery)
Campeonato	getCampeonato()
java.lang.Integer	getIdInscripcion()
Nadadore	getNadadore()
Prueba	getPrueba()
java.util.List< Sery >	getSeries()
java.lang.String	getTiempoPrueba()
Sery	removeSery(Sery sery)
void	setCampeonato(Campeonato campeonato)
void	setIdInscripcion(java.lang.Integer idInscripcion)
void	setNadadore(Nadadore nadadore)
void	setPrueba(Prueba prueba)
void	setSeries(java.util.List<Sery> series)

void | **setTiempoPrueba**(java.lang.String tiempoPrueba)

Methods inherited from class java.lang.Object
--

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

CLASS LOGGIN

java.lang.Object
competencia.modelo.Loggin

All Implemented Interfaces:

java.io.Serializable

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

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Loggin()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entreno	addEntreno(Entreno entreno)
java.lang.String	getCargo()
java.lang.String	getClave()
Entrenador	getEntrenador()
java.util.List< Entreno >	getEntrenos()
java.lang.String	getUsuario()
Entreno	removeEntreno(Entreno entreno)

void	setCargo (java.lang.String cargo)
void	setClave (java.lang.String clave)
void	setEntrenador (Entrenador entrenador)
void	setEntrenos (java.util.List<Entreno> entrenos)
void	setUsuario (java.lang.String usuario)

Methods inherited from class java.lang.Object
--

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

CLASS NADADORE

java.lang.Object
 competencia.modelo.Nadadore

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Nadadore
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description

<u>Nadadore()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
--------------------	----------------------

Inscripcion	addInscripcion (Inscripcion inscripcion)
java.lang.String	getApellidos ()
Categoria	getCategoria ()

java.lang.String	getCiNadador()
Club	getClub()
java.lang.String	getDireccion()
java.lang.Integer	getEdad()
Genero	getGenero()
java.util.List< Inscripcion >	getInscripciones()
java.lang.String	getNombres()
java.lang.String	getTelefono()
Inscripcion	removeInscripcion(Inscripcion inscripcion)
void	setApellidos (java.lang.String apellidos)
void	setCategoria (Categoria categoria)
void	setCiNadador (java.lang.String ciNadador)
void	setClub (Club club)
void	setDireccion (java.lang.String direccion)
void	setEdad (java.lang.Integer edad)
void	setGenero (Genero genero)
void	setInscripciones (java.util.List< Inscripcion > inscripciones)
void	setNombres (java.lang.String nombres)
void	setTelefono (java.lang.String telefono)

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

CLASS NIVELE

java.lang.Object
 competencia.modelo.Nivele

All Implemented Interfaces:

java.io.Serializable


```
@Entity
public class Nivele
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Nivele()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entreno	addEntreno(Entreno entreno)
java.lang.String	getDescripcion()
java.util.List< Entreno >	getEntrenos()
java.lang.Integer	getIdNivel()
Entreno	removeEntreno(Entreno entreno)
void	setDescripcion(java.lang.String descripcion)
void	setEntrenos(java.util.List<Entreno> entrenos)
void	setIdNivel(java.lang.Integer idNivel)

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

CLASS POSICIONE

```
java.lang.Object
  competencia.modelo.Posicione
```

All Implemented Interfaces:

```
java.io.Serializable
```

```
@Entity
public class Posicione
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Posicione()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Campeonato	getCampeonato()
java.lang.String	getCategoria()
java.lang.Integer	getIdPosicion()
java.lang.Integer	getPuesto()
Sery	getSery()
java.lang.String	getTiempo()
void	setCampeonato(Campeonato campeonato)
void	setCategoria(java.lang.String categoria)
void	setIdPosicion(java.lang.Integer idPosicion)
void	setPuesto(java.lang.Integer puesto)
void	setSery(Sery sery)
void	setTiempo(java.lang.String tiempo)

Methods inherited from class java.lang.Object

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

CLASS PRUEBA

```
java.lang.Object
competencia.modelo.Prueba
```

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Prueba
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Prueba()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Inscripcion	addInscripcion(Inscripcion inscripcion)
Tomatiempo	addTomatiempo(Tomatiempo tomatiempo)
java.lang.String	getDescripcion()
java.lang.Integer	getIdPrueba()
java.util.List< Inscripcion >	getInscripciones()
java.util.List< Tomatiempo >	getTomatiempos()
Inscripcion	removeInscripcion(Inscripcion inscripcion)
Tomatiempo	removeTomatiempo(Tomatiempo tomatiempo)
void	setDescripcion(java.lang.String descripcion)
void	setIdPrueba(java.lang.Integer idPrueba)
void	setInscripciones(java.util.List<Inscripcion> inscripciones)
void	setTomatiempos(java.util.List<Tomatiempo> tomatiempos)

Methods inherited from class java.lang.Object

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

CLASS SERY

java.lang.Object
competencia.modelo.Sery

All Implemented Interfaces:

java.io.Serializable

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

RESUMEN DE CONSTRUCTORES

Constructor and Description

<u>Sery()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
--------------------	----------------------

Posicione	addPosicione(Posicione posicione)
Campeonato	getCampeonato()
java.lang.Integer	getCarril()
java.lang.String	getCategoria()
java.lang.Integer	getDescripcion()
java.lang.Integer	getIdSerie()
Inscripcion	getInscripcion()
java.util.List< Posicione >	getPosiciones()
java.lang.String	getPrueba()
java.lang.String	getTiempoFinal()

java.lang.String	getTiempoInicial()
Posicione	removePosicione(Posicione posicion)
void	setCampeonato(Campeonato campeonato)
void	setCarril(java.lang.Integer carril)
void	setCategoria(java.lang.String categoria)
void	setDescription(java.lang.Integer descripcion)
void	setIdSerie(java.lang.Integer idSerie)
void	setInscripcion(Inscripcion inscripcion)
void	setPosiciones(java.util.List<Posicione> posiciones)
void	setPrueba(java.lang.String prueba)
void	setTiempoFinal(java.lang.String tiempoFinal)
void	setTiempoInicial(java.lang.String tiempoInicial)

Methods inherited from class java.lang.Object
--

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

CLASS TEMPORADA

java.lang.Object
 competencia.modelo.Temporada

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Temporada
extends java.lang.Object
implements java.io.Serializable
```

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Temporada()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entreno	addEntreno(Entreno entreno)
java.lang.String	getDescripcion()
java.util.List< Entreno >	getEntrenos()
java.util.Date	getFechaFin()
java.util.Date	getFechaInicio()
java.lang.Integer	getIdTemporada()
java.lang.String	getObjetivo()
Entreno	removeEntreno(Entreno entreno)
void	setDescripcion(java.lang.String descripcion)
void	setEntrenos(java.util.List<Entreno> entrenos)
void	setFechaFin(java.util.Date fechaFin)
void	setFechaInicio(java.util.Date fechaInicio)
void	setIdTemporada(java.lang.Integer idTemporada)
void	setObjetivo(java.lang.String objetivo)

Methods inherited from class java.lang.Object
--

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

CLASS TIPOENTRENAMIENTO

java.lang.Object
competencia.modelo.Tipoentrenamiento

All Implemented Interfaces:

java.io.Serializable

```
@Entity
public class Tipoentrenamiento
extends java.lang.Object
```

implements java.io.Serializable

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Tipoentrenamiento()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Entreno	addEntreno(Entreno entreno)
java.lang.String	getDescripcion()
java.util.List< Entreno >	getEntrenos()
java.lang.Integer	getIdTipoEntrenamiento()
Entreno	removeEntreno(Entreno entreno)
void	setDescripcion(java.lang.String descripcion)
void	setEntrenos(java.util.List<Entreno> entrenos)
void	setIdTipoEntrenamiento(java.lang.Integer idTipoEntrenamiento)

Methods inherited from class java.lang.Object

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

CLASS TOMATIEMPO

```
java.lang.Object
  competencia.modelo.Tomatiempo
```

All Implemented Interfaces:

```
java.io.Serializable
```

```
@Entity
public class Tomatiempo
extends java.lang.Object
```

implements java.io.Serializable

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>Tomatiempo()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
Deportisa	getDeportisa()
java.util.Date	getFecha()
java.lang.Integer	getIdtoma()
Prueba	getPrueba()
java.lang.String	getTiempo()
void	setDeportisa(Deportisa deportisa)
void	setFecha(java.util.Date fecha)
void	setIdtoma(java.lang.Integer idtoma)
void	setPrueba(Prueba prueba)
void	setTiempo(java.lang.String tiempo)

Methods inherited from class java.lang.Object

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

PACKAGES COMPETENCIA.MODELO.MANAGER

En este paquete se detallan las clases y métodos que se utilizan para realizar los diferentes CRUDS del sistema.

CLASS MANAGERCOMPETENCIA

```
java.lang.Object  
  competencia.modelo.manager.ManagerCompetencia
```

```
public class ManagerCompetencia  
  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.

Constructor Detail

ManagerCompetencia

```
public ManagerCompetencia()  
Constructor de la clase ManagerCompetencia. Se encarga de crear los objetos factory y entity manager utilizando el patrón de diseño singleton.
```

Method Detail

finAllClub

```
public java.util.List<Club> finAllClub()  
Método finder para la consulta de clubs. Hace uso del componente ManagerDAO de la capa model.
```

Returns:

listado de clubs.

findClubById

```
public Club findClubById(java.lang.Integer idClub) throws java.lang.Exception
```

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

Parameters:

idClub - código del club que se desea buscar.

Returns:

el club encontrado.

Throws:

java.lang.Exception

insertarClubs

public void insertarClubs(Club c) throws java.lang.Exception

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

Parameters:

c - El nuevo club.

Throws:

java.lang.Exception

eliminarClub

public void eliminarClub(java.lang.Integer idClub) throws java.lang.Exception

Borra de la base de datos un club específico. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idClub - el código del club que se desea eliminar.

Throws:

java.lang.Exception

actualizarClub

public void actualizarClub(Club c) throws java.lang.Exception

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

Parameters:

id_club - identificador del club.

c - Los datos del club que se desea actualizar.

Throws:

java.lang.Exception

validadorDeCédula

public boolean validadorDeCédula(java.lang.String cédula)

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

Parameters:

cédula - cédula a comprobar.

Returns:

boolean true o false.

findAllNadadore

public java.util.List<Nadadore> findAllNadadore()

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

Returns:

listado de nadadores.

findNadadoreById

public Nadadore findNadadoreById(java.lang.String cédula) throws java.lang.Exception

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

Parameters:

cédula - cédula del nadador que se desea buscar.

Returns:

el nadador encontrado.

Throws:

java.lang.Exception

insertarNadador

public void insertarNadador(Nadadore n) throws java.lang.Exception

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

Parameters:

cédula - cédula del nadador que se inserta a la base

n - El nuevo nadador.

Throws:

java.lang.Exception

eliminarNadador

public void eliminarNadador(java.lang.String cédula) throws java.lang.Exception
Borra de la base de datos un nadador específico. Hace uso del componente ManagerDAO de la capa model.

Parameters:

cédula - la cédula del nadador que se desea eliminar

Throws:

java.lang.Exception

actualizarNadador

public void actualizarNadador(Nadadore nadador) throws java.lang.Exception
Actualiza la información de un nadador en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

nadador - Los datos del nadador que se desea actualizar.

Throws:

java.lang.Exception

findAllInscripcion

public java.util.List<Inscripcion> findAllInscripcion()
Método finder para la consulta de parámetros. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de inscripciones.

findInscripcionById

public Inscripcion findInscripcionById(java.lang.Integer id_inscripcion) throws
java.lang.Exception
Método finder para consulta de inscripciones. Hace uso del componente ManagerDAO de la capa model.

Parameters:

id_inscripción - código de la inscripción

Returns:

la inscripción encontrado.

Throws:

java.lang.Exception

InsertarInscripcion

```
public void InsertarInscripcion(Inscripcion i) throws java.lang.Exception
```

Guarda una nueva inscripción en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

id_inscripción - código de la inscripcion

i - La nueva Inscripcion

Throws:

java.lang.Exception

eliminarInscripcion

```
public void eliminarInscripcion(java.lang.Integer idInscripcion) throws java.lang.Exception
```

Borra de la base de datos una inscripción específica. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idInscripción - el código de la inscripción a eliminar

Throws:

java.lang.Exception

actualizarInscripcion

```
public void actualizarInscripcion(Inscripcion inscripcion) throws java.lang.Exception
```

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

Parameters:

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

Throws:

java.lang.Exception

findAllGéneros

```
public java.util.List<Genero> findAllGéneros()
```

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

Returns:

listado de géneros.

findGeneroById

```
public Genero findGeneroById(java.lang.Integer idGenero) throws java.lang.Exception
```

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

Parameters:

idGenero - código del genero

Returns:

el género encontrado.

Throws:

java.lang.Exception

findAllCategorías

```
public java.util.List<Categoria> findAllCategorías()
```

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

Returns:

listado de géneros

findCategoriaById

```
public Categoria findCategoriaById(java.lang.Integer idCategoria) throws java.lang.Exception
```

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

Parameters:

idGenero - código del genero

Returns:

el genero encontrado.

Throws:

java.lang.Exception

findAllUsuarios

```
public java.util.List<Loggin> findAllUsuarios()
```

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

Returns:

listado de usuarios.

findUsuarioById

```
public Loggin findUsuarioById(java.lang.String usuario) throws java.lang.Exception
```

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

Parameters:

usuario - nombre de usuario a encontrar

Returns:

el usuario encontrado.

Throws:

java.lang.Exception

eliminarUsuario

```
public void eliminarUsuario(java.lang.String user) throws java.lang.Exception
```

Borra de la base de datos una inscripción específica. Hace uso del componente ManagerDAO de la capa model.

Parameters:

user - el código del usuario a eliminar

Throws:

java.lang.Exception

insertarUsuario

```
public void insertarUsuario(Loggin u) throws java.lang.Exception
```

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

Parameters:

cédula – cédula del nadador que se inserta a la base
u - El nuevo usuario.

Throws:

java.lang.Exception

actualizarUsuario

public void actualizarUsuario(Loggin usuario) throws java.lang.Exception
Actualiza la información de un usuario en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

usuario - Los datos del usuario que se desea actualizar.

Throws:

java.lang.Exception

findAllPruebas

public java.util.List<Prueba> findAllPruebas()
Método finder para la consulta de pruebas. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de pruebas.

findPruebaById

public Prueba findPruebaById(java.lang.Integer idPrueba) throws java.lang.Exception
Método finder para la consulta de pruebas. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idPrueba - id de prueba a encontrar

Returns:

la prueba encontrado.

Throws:

java.lang.Exception

findAllEstilos

```
public java.util.List<Estilo> findAllEstilos()
```

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

Returns:

listado de estilos.

findEstiloById

```
public Estilo findEstiloById(java.lang.Integer idEstilo) throws java.lang.Exception
```

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

Parameters:

idEstilo - id de estilo a encontrar

Returns:

el estilo encontrado.

Throws:

java.lang.Exception

findAllSeries

```
public java.util.List<Sery> findAllSeries()
```

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

Returns:

listado de series.

findSerieById

```
public Sery findSerieById(java.lang.Integer idSerie) throws java.lang.Exception
```

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

Parameters:

idSerie - id de serie a encontrar

Returns:

la serie encontrado.

Throws:

java.lang.Exception

InsertarSerie

public void InsertarSerie(Sery ser) throws java.lang.Exception

Guarda una nueva serie en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

id_serie - código de la serie

ser - La nueva serie

Throws:

java.lang.Exception

actualizarSerie

public void actualizarSerie(Sery serie) throws java.lang.Exception

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

Parameters:

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

Throws:

java.lang.Exception

findAllPosiciones

public java.util.List<Posicione> findAllPosiciones()

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

Returns:

listado de posiciones.

InsertarPosicion

public void InsertarPosicion(Posicione pos) throws java.lang.Exception

Guarda una nueva serie en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

id_serie - código de la serie

ser - La nueva serie

Throws:

java.lang.Exception

eliminarPosiciones

```
public void eliminarPosiciones(java.lang.Integer id) throws java.lang.Exception
```

Throws:

java.lang.Exception

finAllCampeonatos

```
public java.util.List<Campeonato> finAllCampeonatos()
```

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

Returns:

listado de campeonatos.

findCampeonatoById

```
public Campeonato findCampeonatoById(java.lang.Integer idCampeonato) throws  
java.lang.Exception
```

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

Parameters:

idCampeonato - código del Campeonato que se desea buscar.

Returns:

el Campeonato encontrado.

Throws:

java.lang.Exception

insertarCampeonato

```
public void insertarCampeonato(Campeonato c) throws java.lang.Exception
```

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

Parameters:

c - El nuevo Campeonato.

Throws:

java.lang.Exception

eliminarCampeonato

public void eliminarCampeonato(java.lang.Integer idCampeonato) throws java.lang.Exception
Borra de la base de datos un Campeonato específico. Hace uso del componente ManagerDAO de la capa model.

Parameters:

idClub - el código del Campeonato que se desea eliminar.

Throws:

java.lang.Exception

actualizarCampeonato

public void actualizarCampeonato(Campeonato c) throws java.lang.Exception
Actualiza la información de un club en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

id_club - identificador del club.

c - Los datos del club que se desea actualizar.

Throws:

java.lang.Exception

CLASS MANAGERDAO

java.lang.Object
competencia.modelo.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.

Constructor Detail

ManagerDAO

```
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 nombreMétodo,  
java.lang.String mensaje)
```

Método básico para mostrar mensajes de depuración.

Parameters:

- clase - Información de la clase (Class) para generar el mensaje de depuración.
- nombreMétodo - 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.código,o.nombre

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

Returns:

Listado resultante.

findAll

```
public java.util.List findAll(java.lang.Class clase)
finder Genérico que devuelve todos 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 clausulas 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:

1. o.nombre='Antonio'
2. o.nombre='Antonio' and o.telefono='0444-434'
3. 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.código,o.nombre

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

Returns:

Listado resultante.

findJPQL

```
public java.util.List findJPQL(java.lang.String pClausulaJPQL)
```

Finder Genérico que permite ejecutar cualquier sentencia JPQL.

Parameters:

pClausulaJPQL - Sentencia JPQL que se va a ejecutar.

Returns:

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 MANAGERENTRENAMIENTO

```
java.lang.Object  
competencia.modelo.manager.ManagerEntrenamiento
```

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

Constructor Detail

ManagerEntrenamiento

```
public ManagerEntrenamiento()
```


Method Detail

validadorDeCédula

```
public boolean validadorDeCédula(java.lang.String cédula)
```

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

Parameters:

cédula - cédula a comprobar.

Returns:

boolean true o false.

findAllEntrenadores

```
public java.util.List<Entrenador> findAllEntrenadores()
```

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

Returns:

listado de entrenadores.

findEntrenadorById

```
public Entrenador findEntrenadorById(java.lang.String cédula) throws java.lang.Exception
```

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

Parameters:

cédula - cédula del entrenador que se desea buscar.

Returns:

el entrenador encontrado.

Throws:

java.lang.Exception

insertarEntrenador

```
public void insertarEntrenador(Entrenador e) throws java.lang.Exception
```

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

Parameters:

cédula - cédula del entrenador que se inserta a la base

e - El nuevo entrenador.

Throws:

java.lang.Exception

eliminarEntrenador

public void eliminarEntrenador(java.lang.String cédula) throws java.lang.Exception
Borra de la base de datos un entrenador específico. Hace uso del componente ManagerDAO de la capa model.

Parameters:

cédula - la cédula del entrenador que se desea eliminar

Throws:

java.lang.Exception

actualizarEntrenador

public void actualizarEntrenador(Entrenador entrenador) throws java.lang.Exception
Actualiza la información de un entrenador en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

entrenador - Los datos del entrenador que se desea actualizar.

Throws:

java.lang.Exception

findAllDeportista

public java.util.List<Deportista> findAllDeportista()
Método finder para la consulta de deportistas. Hace uso del componente ManagerDAO de la capa model.

Returns:

listado de deportistas.

findDeportistaById

public Deportista findDeportistaById(java.lang.String cédula) throws java.lang.Exception
Método finder para consulta de deportistas. Hace uso del componente ManagerDAO de la capa model.

Parameters:

cédula - cédula del deportista que se desea buscar.

Returns:

el deportista encontrado.

Throws:

java.lang.Exception

insertarDeportista

public void insertarDeportista(Deportista e) throws java.lang.Exception

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

Parameters:

cédula - cédula del deportista que se inserta a la base

e - El nuevo deportista.

Throws:

java.lang.Exception

eliminarDeportista

public void eliminarDeportista(java.lang.String cédula) throws java.lang.Exception

Borra de la base de datos un deportista específico. Hace uso del componente ManagerDAO de la capa model.

Parameters:

cédula - la cédula del deportista que se desea eliminar

Throws:

java.lang.Exception

actualizarDeportista

public void actualizarDeportista(Deportista dep) throws java.lang.Exception

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

Parameters:

dep - Los datos del deportista que se desea actualizar.

Throws:

java.lang.Exception

findAllTomatiempo

```
public java.util.List<Tomatiempo> findAllTomatiempo()
```

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

Returns:

listado de tomaTiempos.

InsertarTomatiempo

```
public void InsertarTomatiempo(Tomatiempo t) throws java.lang.Exception
```

Guarda una nueva serie en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

t - nueva toma de tiempos

Throws:

java.lang.Exception

findAllEntrenamientodiario

```
public java.util.List<Entrenamientodiario> findAllEntrenamientodiario()
```

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

Returns:

listado de entrenamientos diarios

InsertarEntrenamientodiario

```
public void InsertarEntrenamientodiario(Entrenamientodiario e) throws java.lang.Exception
```

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

Parameters:

e - nuevo Entrenamientodiario

Throws:

java.lang.Exception

FindAllEntrenamientos

```
public java.util.List<Entreno> findAllEntrenamientos()
```

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

Returns:

listado de entrenamientos.

InsertarEntrenamiento

```
public void InsertarEntrenamiento(Entreno e) throws java.lang.Exception
```

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

Parameters:

e - nuevo Entrenamientodiario

Throws:

java.lang.Exception

FindAllHorarios

```
public java.util.List<Horario> findAllHorarios()
```

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

Returns:

listado de Niveles.

FindHorarioById

```
public Horario findHorarioById(java.lang.Integer cod) throws java.lang.Exception
```

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

Parameters:

cod - cod del horario que se desea buscar.

Returns:

el horario encontrado.

Throws:

java.lang.Exception

findAllNiveles

```
public java.util.List<Nivele> findAllNiveles()
```

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

Returns:

listado de Niveles.

FindNivelById

```
public Nivele findNivelById(java.lang.Integer cod) throws java.lang.Exception
```

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

Parameters:

cod - cod del nivel que se desea buscar.

Returns:

el nivel encontrado.

Throws:

java.lang.Exception

FindAlltipoEntre

```
public java.util.List<Tipoentrenamiento> findAlltipoEntre()
```

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

Returns:

listado de Tipos de entrenamientos.

FindtipoEntrelById

```
public Tipoentrenamiento findtipoEntrelById(java.lang.Integer cod) throws java.lang.Exception
```

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

Parameters:

cod - cod del Tipos de entrenamientos que se desea buscar.

Returns:

el Tipos de entrenamientos encontrado.

Throws:

java.lang.Exception

FindAllTemporadas

```
public java.util.List<Temporada> findAllTemporadas()
```

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

Returns:

listado de temporadas.

FindTemporadaById

```
public Temporada findTemporadaById(java.lang.Integer código) throws java.lang.Exception
```

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

Parameters:

código - código de la temporada que se desea buscar.

Returns:

la temporada encontrada.

Throws:

java.lang.Exception

InsertarTemporada

```
public void insertarTemporada(Temporada t) throws java.lang.Exception
```

Guarda una nueva Temporada en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

e - El nuevo entrenador.

Throws:

java.lang.Exception

eliminarTemporada

```
public void eliminarTemporada(Temporada código) throws java.lang.Exception
```

Borra de la base de datos una Temporada específico. Hace uso del componente ManagerDAO de la capa model.

Parameters:

código - el código de la Temporada que se desea eliminar

Throws:

java.lang.Exception

ActualizarTemporada

public void actualizarTemporada(Temporada temporada) throws java.lang.Exception
 Actualiza la información de un entrenador en la base de datos. Hace uso del componente ManagerDAO de la capa model.

Parameters:

temporada - Los datos de la Temporada que se desea actualizar.

Throws:

java.lang.Exception

PACKAGES COMPETENCIA.CONTROLLER

En este paquete se detallan las clases y métodos que se usarán para realizar las diferentes operaciones en el sistema.

CLASS BEANAVANCES

java.lang.Object
 competencia.controller.BeanAvances

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

Clase que permite crear, actualizar, visualizar, eliminar a un deportista del club

RESUMEN DE CONSTRUCTORES

Constructor and Description
BeanAvances()

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
<i>java.lang.String</i>	<u>actionInsertarEntrenaDiario()</u> Método que permite el ingreso de un nuevo Entrenamiento diario.
<i>java.lang.String</i>	<u>actionInsertarTomatiempo()</u> Método que permite el ingreso de una nueva toma de tiempos
<i>java.lang.String</i>	<u>cargar()</u>
<i>java.lang.String</i>	<u>getApellidos()</u>
<i>java.lang.String</i>	<u>getCedulaDepor()</u>
<i>static java.lang.String</i>	<u>getCiDepor()</u>
<i>java.lang.String</i>	<u>getCiDeportista()</u>
<u>Deportisa</u>	<u>getDepor()</u>
<i>java.lang.String</i>	<u>getDeportista()</u> Método que permite la carga de datos de un deportista para la visualización de sus avances
<i>java.lang.String</i>	<u>getDireccion()</u>
<i>java.lang.Integer</i>	<u>getEdad()</u>
<i>java.util.Date</i>	<u>getFecha()</u>
<i>java.util.Date</i>	<u>getFechaEntrena()</u>
<i>java.util.Date</i>	<u>getFechaNac()</u>
<i>java.lang.String</i>	<u>getFoto()</u>
<i>java.lang.Integer</i>	<u>getHorario()</u>
<i>java.lang.String</i>	<u>getIdCategoria()</u>
<i>java.lang.Integer</i>	<u>getIdentrenamiento()</u>
<i>java.lang.String</i>	<u>getIdEstilo()</u>
<i>java.lang.Integer</i>	<u>getIdPrueba()</u>
<i>java.lang.Integer</i>	<u>getIdtoma()</u>
<i>java.util.List<<u>Deportisa</u>></i>	<u>getListaDeportista()</u> Devuelve un listado de deportistas SelectItem a partir de un listado de Deportista.
<i>java.util.List<<u>Entrenamientodiario</u>></i>	<u>getListaEntrenamientodiario()</u> Devuelve un listado de todos los entrenamientos acudidos por un deportista
<i>java.util.List<<u>Horario</u>></i>	<u>getlistaHorarios()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Pruebas.
<i>java.util.List<<u>Prueba</u>></i>	<u>getListaPruebas()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Pruebas.
<i>java.util.List<<u>Tomatiempo</u>></i>	<u>getListaTomatiempos()</u> Devuelve un listado de todas las pruebas realizadas por un deportista
<i>java.lang.String</i>	<u>getLugar()</u>
<i>java.lang.String</i>	<u>getNombres()</u>

<i>java.lang.String</i>	<u>getTelefono()</u>
<i>java.lang.String</i>	<u>getTiempo()</u>
<i>static java.lang.String</i>	<u>getUsua()</u>
<i>void</i>	<u>setApellidos</u> (java.lang.String apellidos)
<i>void</i>	<u>setCedulaDepor</u> (java.lang.String cedulaDepor)
<i>static void</i>	<u>setCiDepor</u> (java.lang.String ciDepor)
<i>void</i>	<u>setCiDeportista</u> (java.lang.String ciDeportista)
<i>void</i>	<u>setDepor</u> (Deportisa depor)
<i>void</i>	<u>setDireccion</u> (java.lang.String direccion)
<i>void</i>	<u>setEdad</u> (java.lang.Integer edad)
<i>void</i>	<u>setFecha</u> (java.util.Date fecha)
<i>void</i>	<u>setFechaEntrena</u> (java.util.Date fechaEntrena)
<i>void</i>	<u>setFechaNac</u> (java.util.Date fechaNac)
<i>void</i>	<u>setFoto</u> (java.lang.String foto)
<i>void</i>	<u>setHorario</u> (java.lang.Integer horario)
<i>void</i>	<u>setIdCategoria</u> (java.lang.String idCategoria)
<i>void</i>	<u>setIdEntrenamiento</u> (java.lang.Integer identrenamiento)
<i>void</i>	<u>setIdEstilo</u> (java.lang.String idEstilo)
<i>void</i>	<u>setIdPrueba</u> (java.lang.Integer idPrueba)
<i>void</i>	<u>setIdtoma</u> (java.lang.Integer idtoma)
<i>void</i>	<u>setLugar</u> (java.lang.String lugar)
<i>void</i>	<u>setNombres</u> (java.lang.String nombres)
<i>void</i>	<u>setTelefono</u> (java.lang.String telefono)
<i>void</i>	<u>setTiempo</u> (java.lang.String tiempo)
<i>static void</i>	<u>setUsua</u> (java.lang.String usua)

CLASS BEANCAMPEONATO

java.lang.Object
 competencia.controller.BeanCampeonato

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

Clase que permite crear, actualizar, visualizar, eliminar a un club de la competencia

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanCampeonato()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>actionActualizarCampeonato()</u> Método que permite actualizar los valores de un campeonato
<i>java.lang.String</i>	<u>actionCargarCampeonato(Campeonato c)</u> Método que permite cargar los datos del campeonato para poder actualizarlos
<i>java.lang.String</i>	<u>actionEliminarCampeonato(Campeonato campeonato)</u>
<i>java.lang.String</i>	<u>actionInsertarCampeonato()</u> Método que permite el ingreso de un nuevo campeonato
<i>java.lang.String</i>	<u>cancelarCampeonato()</u>
<i>java.lang.String</i>	<u>getEstado()</u>
<i>java.util.Date</i>	<u>getFechafin()</u>
<i>java.util.Date</i>	<u>getFechainicio()</u>
<i>java.lang.Integer</i>	<u>getIdcampeonato()</u>
<i>java.util.List<Campeonato></i>	<u>getListaCampeonatos()</u> Devuelve un listado de todos los campeonats
<i>java.lang.String</i>	<u>getNombre()</u>
<i>void</i>	<u>setEstado(java.lang.String estado)</u>
<i>void</i>	<u>setFechafin(java.util.Date fechafin)</u>
<i>void</i>	<u>setFechainicio(java.util.Date fechainicio)</u>
<i>void</i>	<u>setIdcampeonato(java.lang.Integer idcampeonato)</u>
<i>void</i>	<u>setNombre(java.lang.String nombre)</u>

CLASS BEANCLUB

```
java.lang.Object
    competencia.controller.BeanClub
```

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

Clase que permite crear, actualizar, visualizar, eliminar a un club de la competencia

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanClub()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
---------------------------	-----------------------------

<i>java.lang.String</i>	<u>actionActualizarClub()</u> Método que permite actualizar los datos de un club
<i>java.lang.String</i>	<u>actionCargarClub(Club club)</u> Método que permite cargar los datos de un club seleccionado para actualizar sus valores
<i>java.lang.String</i>	<u>actionEliminarClub(Club club)</u>
<i>java.lang.String</i>	<u>actionInsertarClub()</u> Método que permite el ingreso de un nuevo club
<i>java.lang.String</i>	<u>cancelarClub()</u>
<i>java.lang.String</i>	<u>getCiudad()</u>
<i>java.lang.String</i>	<u>getCorreo()</u>
<i>java.lang.String</i>	<u>getEntrenador()</u>
<i>java.lang.Integer</i>	<u>getIdclub()</u>
<i>java.util.List<Club></i>	<u>getListaClubs()</u>
<i>java.lang.String</i>	<u>getNombre()</u>
<i>java.lang.String</i>	<u>getTelefono()</u>
<i>void</i>	<u>setCiudad</u> (java.lang.String ciudad)
<i>void</i>	<u>setCorreo</u> (java.lang.String correo)
<i>void</i>	<u>setEntrenador</u> (java.lang.String entrenador)
<i>void</i>	<u>setIdclub</u> (java.lang.Integer idclub)
<i>void</i>	<u>setNombre</u> (java.lang.String nombre)
<i>void</i>	<u>setTelefono</u> (java.lang.String telefono)

CLASS BEANDEPORTISTA

java.lang.Object
competencia.controller.BeanDeportista

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

Clase que permite crear, actualizar, visualizar, eliminar a un deportista del club

RESUMEN DE CONSTRUCTORES

Constructor and Description
--

BeanDeportista()

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>actionActualizarDeportista()</u> Método que permite la actualización de un deportista
<i>java.lang.String</i>	<u>actionCargarDeportista(Deportista d)</u> Método que permite cargar los datos de un deportista para la actualización de sus datos
<i>java.lang.String</i>	<u>actionEliminarDeportista(Deportista nad)</u> Método que permite eliminar un deportista
<i>java.lang.String</i>	<u>actionInsertarDeportista()</u> Método que permite el ingreso de un nuevo deportista
<i>static java.lang.Integer</i>	<u>calcularEdad(java.util.Date fecha)</u> Método que permite calcular la edad
<i>java.lang.String</i>	<u>cancelarDeportista()</u>
<i>java.lang.String</i>	<u>getApellidos()</u>
<i>java.lang.String</i>	<u>getCedula Depor()</u>
<i>java.lang.String</i>	<u>getDestination()</u>
<i>java.lang.String</i>	<u>getDireccion()</u>
<i>java.lang.Integer</i>	<u>getEdad()</u>
<i>java.util.Date</i>	<u>getFechaNac()</u>
<i>org.primefaces.model.UploadedFile</i>	<u>getFile()</u>
<i>java.lang.String</i>	<u>getFoto()</u>
<i>java.lang.Integer</i>	<u>getIdCategoria()</u>
<i>java.lang.Integer</i>	<u>getIdEstilo()</u>
<i>java.lang.Integer</i>	<u>getIdGenero()</u>
<i>java.util.List<Categoria></i>	<u>getListaCategoria()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Categoria.
<i>java.util.List<Deportista></i>	<u>getListaDeportista()</u> Método que permite obtener un listado de los deportistas ingresados en la base
<i>java.util.List<Estilo></i>	<u>getListaEstilo()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Estilo.
<i>java.util.List<Genero></i>	<u>getListaGenero()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Genero.
<i>java.lang.String</i>	<u>getNombreRepresentante()</u>
<i>java.lang.String</i>	<u>getNombres()</u>
<i>java.lang.String</i>	<u>getParentescoRepresentante()</u>
<i>java.lang.String</i>	<u>getTelefono()</u>

<i>java.lang.String</i>	<u>getTelefonoRepresentante()</u>
<i>void</i>	<u>setApellidos</u> (java.lang.String apellidos)
<i>void</i>	<u>setCedulaDepor</u> (java.lang.String cedulaDepor)
<i>void</i>	<u>setDestinacion</u> (java.lang.String destination)
<i>void</i>	<u>setDireccion</u> (java.lang.String direccion)
<i>void</i>	<u>setEdad</u> (java.lang.Integer edad)
<i>void</i>	<u>setFechaNac</u> (java.util.Date fechaNac)
<i>void</i>	<u>setFile</u> (org.primefaces.model.UploadedFile file)
<i>void</i>	<u>setFoto</u> (java.lang.String foto)
<i>void</i>	<u>setIdCategoria</u> (java.lang.Integer idCategoria)
<i>void</i>	<u>setIdEstilo</u> (java.lang.Integer idEstilo)
<i>void</i>	<u>setIdGenero</u> (java.lang.Integer idGenero)
<i>void</i>	<u>setNombreRepresentante</u> (java.lang.String nombreRepresentante)
<i>void</i>	<u>setNombres</u> (java.lang.String nombres)
<i>void</i>	<u>setParentescoRepresentante</u> (java.lang.String parentescoRepresentante)
<i>void</i>	<u>setTelefono</u> (java.lang.String telefono)
<i>void</i>	<u>setTelefonoRepresentante</u> (java.lang.String telefonoRepresentante)
<i>void</i>	<u>upload()</u> Método que permite cargar la foto de un deportista duplicando el carchi en la carpeta del proyecto

CLASS BEANENTRENADOR

```
java.lang.Object
    competencia.controller.BeanEntrenador
```

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

Clase que permite crear, actualizar, visualizar, eliminar a un entrenador

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanEntrenador()</u>

RESUMEN DE MÉTODOS

MODIFICADOR Y TIPO	MÉTODO Y DESCRIPCIÓN
<i>java.lang.String</i>	<u>actionActualizarEntrenador()</u>
<i>java.lang.String</i>	<u>actionCargarEntrenador(Entrenador entre)</u>
<i>java.lang.String</i>	<u>actionEliminarNadador(Entrenador entre)</u>
<i>java.lang.String</i>	<u>actionInsertarEntrenador()</u>
<i>static java.lang.Integer</i>	<u>calcularEdad(java.util.Date fecha)</u> Método que permite calcular la edad del entrenador
<i>java.lang.String</i>	<u>cancelarEntrenador()</u>
<i>java.lang.String</i>	<u>getApellidos()</u>
<i>java.lang.String</i>	<u>getCedulaEntre()</u>
<i>java.lang.String</i>	<u>getClub()</u>
<i>java.lang.String</i>	<u>getDestination()</u>
<i>java.lang.String</i>	<u>getDireccion()</u>
<i>java.lang.Integer</i>	<u>getEdad()</u>
<i>java.util.Date</i>	<u>getFechaNac()</u>
<i>org.primefaces.model.UploadedFile</i>	<u>getFile()</u>
<i>java.lang.String</i>	<u>getFoto()</u>
<i>java.lang.Integer</i>	<u>getIdGenero()</u>
<i>java.util.List<Entrenador></i>	<u>getListaEntrenadores()</u>
<i>java.util.List<Genero></i>	<u>getListaGenero()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Genero.
<i>java.lang.String</i>	<u>getNombres()</u>
<i>java.lang.String</i>	<u>getTelefono()</u>
<i>void</i>	<u>setApellidos(java.lang.String apellidos)</u>
<i>void</i>	<u>setCedulaEntre(java.lang.String cedulaEntre)</u>
<i>void</i>	<u>setClub(java.lang.String club)</u>
<i>void</i>	<u>setDestination(java.lang.String destination)</u>
<i>void</i>	<u>setDireccion(java.lang.String direccion)</u>
<i>void</i>	<u>setEdad(java.lang.Integer edad)</u>
<i>void</i>	<u>setFechaNac(java.util.Date fechaNac)</u>
<i>void</i>	<u>setFile(org.primefaces.model.UploadedFile file)</u>
<i>void</i>	<u>setFoto(java.lang.String foto)</u>
<i>void</i>	<u>setIdGenero(java.lang.Integer idGenero)</u>
<i>void</i>	<u>setNombres(java.lang.String nombres)</u>
<i>void</i>	<u>setTelefono(java.lang.String telefono)</u>
<i>void</i>	<u>upload()</u> Método que permite cargar una imagen al proyecto.

CLASS BEANENTRENAMIENTO

java.lang.Object
competencia.controller.BeanEntrenamiento

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

Clase que permite la creación y almacenamiento de una entrenamiento.

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanEntrenamiento()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>addEvent</u> (javax.faces.event.ActionEvent actionEvent) Método que permite insertar un nuevo entrenamietno
<i>java.lang.Integer</i>	<u>getCodnivel()</u>
<i>java.lang.Integer</i>	<u>getCodtipoEntrenamiento()</u>
<i>java.lang.String</i>	<u>getDescripcion()</u>
<i>java.lang.Integer</i>	<u>getDistancia()</u>
<i>org.primefaces.model.ScheduleEvent</i>	<u>getEvent()</u>
<i>org.primefaces.model.ScheduleModel</i>	<u>getEventModel()</u>
<i>java.lang.Integer</i>	<u>getHorario()</u>
<i>java.lang.Integer</i>	<u>getIdcond()</u>
<i>java.util.Date</i>	<u>getInitialDate()</u>
<i>org.primefaces.model.ScheduleModel</i>	<u>getLazyEventModel()</u>
<i>java.util.List<<u>Horario</u>></i>	<u>getlistaHorarios()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Horarios.
<i>java.util.List<<u>Nivele</u>></i>	<u>getListaNiveles()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Niveles.
<i>java.util.List<<u>Tipoentrenamiento</u>></i>	<u>getListaTipos()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Niveles.
<i>java.util.Date</i>	<u>getRandomDate</u> (java.util.Date base)
<i>static java.lang.String</i>	<u>getUsuario()</u>
<i>void</i>	<u>mostrar()</u>

	Método que permite cargar los entrenamientos del usuario actual.
<i>void</i>	<u>onDateSelect</u> (org.primefaces.event.SelectEvent selectEvent)
<i>void</i>	<u>setCodnivel</u> (java.lang.Integer codnivel)
<i>void</i>	<u>setCodtipoEntrenamiento</u> (java.lang.Integer codtipoEntrenamiento)
<i>void</i>	<u>setDescription</u> (java.lang.String descripcion)
<i>void</i>	<u>setDistancia</u> (java.lang.Integer distancia)
<i>void</i>	<u>setEvent</u> (org.primefaces.model.ScheduleEvent event)
<i>void</i>	<u>setHorario</u> (java.lang.Integer horario)
<i>void</i>	<u>setIdcond</u> (java.lang.Integer idcond)
<i>static void</i>	<u>setUsuario</u> (java.lang.String usuario)

CLASS BEANINSCRIPCIONE

java.lang.Object
competencia.controller.BeanInscripcione

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

Clase que permite crear, actualizar, visualizar, eliminar a un nadador de una prueba

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanInscripcione()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>actionActualizarPrueba()</u>
<i>java.lang.String</i>	<u>actionCargarPrueba</u> (<u>Inscripcion</u> prueba)
<i>java.lang.String</i>	<u>actionEliminarPrueba</u> (<u>Inscripcion</u> ins)
<i>java.lang.String</i>	<u>actionInsertarPrueba()</u>
<i>java.lang.String</i>	<u>cancelarPrueba()</u>

<i>java.lang.String</i>	<u>getCiNadador()</u>
<i>java.lang.Integer</i>	<u>getCodCampeonato()</u>
<i>java.lang.Integer</i>	<u>getIdInscripcion()</u>
<i>java.lang.Integer</i>	<u>getIdPrueba()</u>
<i>java.util.List<<u>Inscripcion</u>></i>	<u>getLisInscripciones()</u> Devuelve un listado de las pruebas del campeonato actual en las que esta inscrito un deportista.
<i>java.util.List<<u>Campeonato</u>></i>	<u>getListaCampeonatos()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Campeonato.
<i>java.util.List<<u>Nadadore</u>></i>	<u>getListaNadadores()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Nadadores.
<i>java.util.List<<u>Prueba</u>></i>	<u>getListaPruebas()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Pruebas.
<i>java.lang.String</i>	<u>getTiempoPrueba()</u>
<i>void</i>	<u>setCiNadador()</u> (<i>java.lang.String ciNadador</i>)
<i>void</i>	<u>setCodCampeonato()</u> (<i>java.lang.Integer codCampeonato</i>)
<i>void</i>	<u>setIdInscripcion()</u> (<i>java.lang.Integer idInscripcion</i>)
<i>void</i>	<u>setIdPrueba()</u> (<i>java.lang.Integer idPrueba</i>)
<i>void</i>	<u>setTiempoPrueba()</u> (<i>java.lang.String tiempoPrueba</i>)

CLASS BEANLOGIN

java.lang.Object
competencia.controller.BeanLogin

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

Clase que permite ingresar al Sistema de Planificacion de Entrenamientos

RESUMEN DE CONSTRUCTORES

Constructor and Description

<u>BeanLogin()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>actionActualizarUsuario()</u>
<i>java.lang.String</i>	<u>actionCargarUsuario(Loggin us)</u>
<i>java.lang.String</i>	<u>actionEliminarUsuario(Loggin entre)</u>
<i>java.lang.String</i>	<u>actionInsertarUsuario()</u>
<i>java.lang.String</i>	<u>cancelarUsuario()</u>
<i>java.lang.String</i>	<u>comprobarIngreso()</u>
	Método que permite al usuario acceder al sistema.
<i>static java.lang.String</i>	<u>encriptar(java.lang.String cadena)</u>
	Encripta un mensaje de texto mediante algoritmo de resumen de mensaje.
<i>java.lang.String</i>	<u>getCargo()</u>
<i>java.lang.String</i>	<u>getCiEntre()</u>
<i>java.lang.String</i>	<u>getClave()</u>
<i>java.util.List<Entrenador></i>	<u>getListaEntrenadores()</u>
	Devuelve un listado de componentes SelectItem a partir de un listado de Entrenador.
<i>java.util.List<Loggin></i>	<u>getListaUsuarios()</u>
	Devuelve un listado de todos los usuarios creados en el sistema.
<i>java.lang.String</i>	<u>getUsuario()</u>
<i>java.lang.String</i>	<u>salirSistema()</u>
	Método que permite salir y cerrar el sistema.
<i>void</i>	<u>setCargo(java.lang.String cargo)</u>
<i>void</i>	<u>setCiEntre(java.lang.String ciEntre)</u>
<i>void</i>	<u>setClave(java.lang.String clave)</u>
<i>void</i>	<u>setUsuario(java.lang.String Usuario)</u>

CLASS BEANNADADORE

```
java.lang.Object
    competencia.controller.BeanNadadore
```

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

Clase que permite crear, actualizar, visualizar, eliminar a un nadador de la competencia

RESUMEN DE CONSTRUCTORES

Constructor and Description

BeanNadadore()

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>actionActualizarNadador()</u>
<i>java.lang.String</i>	<u>actionCargarNadador(Nadadore nadador)</u>
<i>java.lang.String</i>	<u>actionEliminarNadador(Nadadore nad)</u>
<i>java.lang.String</i>	<u>actionInsertarNadador()</u>
<i>java.lang.String</i>	<u>cancelarNadador()</u>
<i>java.lang.String</i>	<u>getApellidos()</u>
<i>java.lang.String</i>	<u>getCiNadador()</u>
<i>java.lang.String</i>	<u>getDireccion()</u>
<i>java.lang.Integer</i>	<u>getEdad()</u>
<i>java.lang.Integer</i>	<u>getIdCategoria()</u>
<i>java.lang.Integer</i>	<u>getIdClub()</u>
<i>java.lang.Integer</i>	<u>getIdgenero()</u>
<i>java.util.List<Categoria></i>	<u>getListaCategoria()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Categoria.
<i>java.util.List<Club></i>	<u>getListaClub()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Clubs.
<i>java.util.List<Genero></i>	<u>getListaGenero()</u> Devuelve un listado de componentes SelectItem a partir de un listado de Genero.
<i>java.util.List<Nadadore></i>	<u>getListaNadadores()</u> Devuelve un listado de todos los nadadores inscritos para una competencia.
<i>java.lang.String</i>	<u>getNombres()</u>
<i>java.lang.String</i>	<u>getTelefono()</u>
<i>void</i>	<u>setApellidos(java.lang.String apellidos)</u>
<i>void</i>	<u>setCiNadador(java.lang.String ciNadador)</u>
<i>void</i>	<u>setDireccion(java.lang.String direccion)</u>
<i>void</i>	<u>setEdad(java.lang.Integer edad)</u>
<i>void</i>	<u>setIdCategoria(java.lang.Integer idCategoria)</u>
<i>void</i>	<u>setIdClub(java.lang.Integer idClub)</u>
<i>void</i>	<u>setIdgenero(java.lang.Integer idgenero)</u>
<i>void</i>	<u>setNombres(java.lang.String nombres)</u>
<i>void</i>	<u>setTelefono(java.lang.String telefono)</u>

CLASS BEANPOSICIONES

```
java.lang.Object  
    competencia.controller.BeanPosiciones
```

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

Clase que permite generar las posiciones

RESUMEN DE CONSTRUCTORES

Constructor and Description

<u>BeanPosiciones()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>void</i>	<u>Estado()</u> Método que permite cambiar el estado de un campeonato de Actual a Finalizado
<i>java.lang.String</i>	<u>generarResultados()</u> Método que permite generar las posiciones a partir de los resultados obtenidos en las pruebas.
<i>java.util.List<<u>Posicione</u>></i>	<u>getListPosicione()</u> Devuelve un listado de las posiciones del campeonato actual.

CLASS BEANREPORTE

```
java.lang.Object  
    competencia.controller.BeanReporte
```

```
public class BeanReporte
```

extends java.lang.Object

Clase que permite la creación de los gráficos estadísticos de las distancias en los entrenamientos y de los resultados obtenidos en una prueba.

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanReporte()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>org.primefaces.model.chart.HorizontalBarChartModel</i>	<u>getBarModel()</u> Método que permte generar la gráfica de los tiempos en las pruebas dadas
<i>org.primefaces.model.chart.BarChartModel</i>	<u>getDistancias()</u> Método que permte generar la gráfica de las distancias nadadas.
<i>void</i>	<u>init()</u>

CLASS BEANSERIES

java.lang.Object
competencia.controller.BeanSeries

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

Clase que permite generar carriles

RESUMEN DE CONSTRUCTORES

Constructor and Description
<u>BeanSeries()</u>

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
----------------------------------	------------------------------------

<i>java.lang.String</i>	<u>actionCargarSerie(Sery serie)</u>
<i>java.lang.String</i>	<u>actionIngresarTiempo()</u>
<i>java.lang.String</i>	<u>cancelarSerie()</u>
<i>java.lang.String</i>	<u>gererarSeries()</u>
	Método que permite generar el número de la serie y el carril en el que se ubicará cada depotista.
<i>java.lang.Integer</i>	<u>getCarril()</u>
<i>java.lang.String</i>	<u>getCategoria()</u>
<i>java.lang.Integer</i>	<u>getCodCampeonato()</u>
<i>java.lang.Integer</i>	<u>getDescripcion()</u>
<i>java.lang.Integer</i>	<u>getIdInscripcion()</u>
<i>java.lang.Integer</i>	<u>getIdSerie()</u>
<i>java.util.List<Sery></i>	<u>getLisSeries()</u>
	Devuelve el listado de todas las series del campeonato actual.
<i>java.lang.String</i>	<u>getPrueba()</u>
<i>java.lang.String</i>	<u>getTiempoFinal()</u>
<i>java.lang.String</i>	<u>getTiempoInicial()</u>
<i>void</i>	<u>setCarril(java.lang.Integer carril)</u>
<i>void</i>	<u>setCategoria(java.lang.String categoria)</u>
<i>void</i>	<u>setCodCampeonato(java.lang.Integer codCampeonato)</u>
<i>void</i>	<u>setDescripcion(java.lang.Integer descripcion)</u>
<i>void</i>	<u>setIdInscripcion(java.lang.Integer idInscripcion)</u>
<i>void</i>	<u>setIdSerie(java.lang.Integer idSerie)</u>
<i>void</i>	<u>setPrueba(java.lang.String prueba)</u>
<i>void</i>	<u>setTiempoFinal(java.lang.String tiempoFinal)</u>
<i>void</i>	<u>setTiempoInicial(java.lang.String tiempoInicial)</u>

CLASS BEANTEMPORADA

java.lang.Object
competencia.controller.BeanTemporada

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

Clase que permite crear, actualizar, visualizar, eliminar a un deportista del club

RESUMEN DE CONSTRUCTORES

Constructor and Description

BeanTemporada()

RESUMEN DE MÉTODOS

<i>MODIFICADOR Y TIPO</i>	<i>MÉTODO Y DESCRIPCIÓN</i>
<i>java.lang.String</i>	<u>actionActualizarTemporada()</u>
<i>java.lang.String</i>	<u>actionCargarTemporada(Temporada d)</u>
<i>java.lang.String</i>	<u>actionInsertarTemporada()</u>
<i>java.lang.String</i>	<u>cancelarTemporada()</u>
<i>java.util.Date</i>	<u>getFecha_fin()</u>
<i>java.util.Date</i>	<u>getFecha_inicio()</u>
<i>java.lang.Integer</i>	<u>getId_temporada()</u>
<i>java.util.List<<u>Temporada</u>></i>	<u>getListaTemporadas()</u> Devuelve un listado de todas las temporadas creaas por un entrenador.
<i>java.lang.String</i>	<u>getNombre()</u>
<i>java.lang.String</i>	<u>getObjetivo()</u>
<i>void</i>	<u>setFecha_fin</u> (java.util.Date fecha_fin)
<i>void</i>	<u>setFecha_inicio</u> (java.util.Date fecha_inicio)
<i>void</i>	<u>setId_temporada</u> (java.lang.Integer id_temporada)
<i>void</i>	<u>setNombre</u> (java.lang.String nombre)
<i>void</i>	<u>setObjetivo</u> (java.lang.String objetivo)

ALOJAMIENTO DEL SISTEMA EN OPENSIFT

1. Se ubica en la página <https://www.openshift.com/app/account/new> en donde se ingresa los datos personales para crear una cuenta.

OPENSIFT

Create an account

Email address
mrcarrillo@utn.edu.ec

Password
.....

Password confirmation
.....

Are you a spam bot?
135

Get Another Get an audio CAPTCHA Help

reCAPTCHA provided by Google - help fight spam and fix books!

By signing up you agree to the [Terms of Service](#) and the [Privacy Policy](#)

Sign Up

Already have an account? [Sign in](#)

You'll love OpenShift because it has:

- Built-in support for Java, Node.js, Ruby, Python, PHP, Perl and extensible functionality to add other languages.
- Powerful command line client tools and a web management console to launch and manage your applications
- Pre-created quickstarts to instantaneously boot your favorite application framework
- A vibrant community backed by an army of developers, evangelists, and OpenShift devotees.
- A wide range of developer resources, including technology specific get started pages, how-to blog posts and videos.

Learn more about [OpenShift](#)

2. Se ingresa al mail y se verifica el mensaje de activación de cuenta y se da clic en Verify Your Account.

Verify your account with OpenShift Online



Thanks for signing up to use OpenShift Online!

Please click the link below to activate your account:

[Verify Your Account](#)

The OpenShift Team

<https://www.openshift.com>

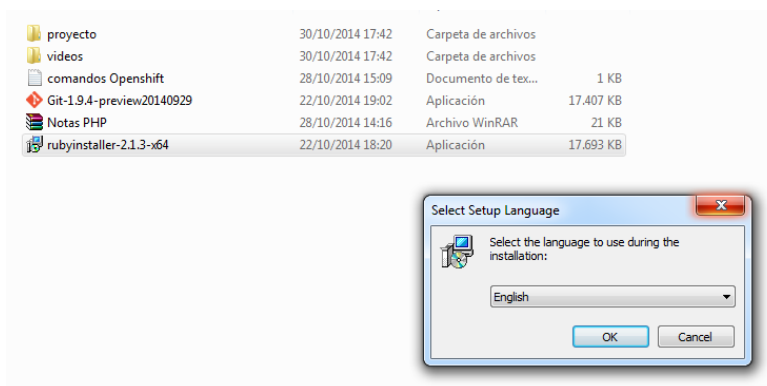
3. Se acepta las condiciones de contrato



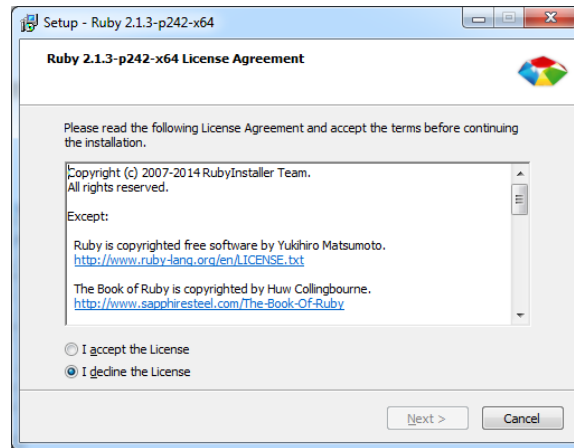
4. Se muestra la ventana de como usar OpenShift paso a paso



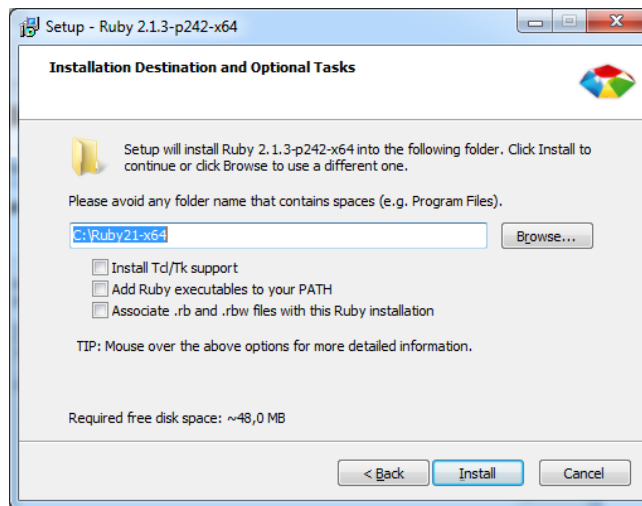
5. Se instala la herramienta Rubí ya que es indispensable para poder alojar el sistema.



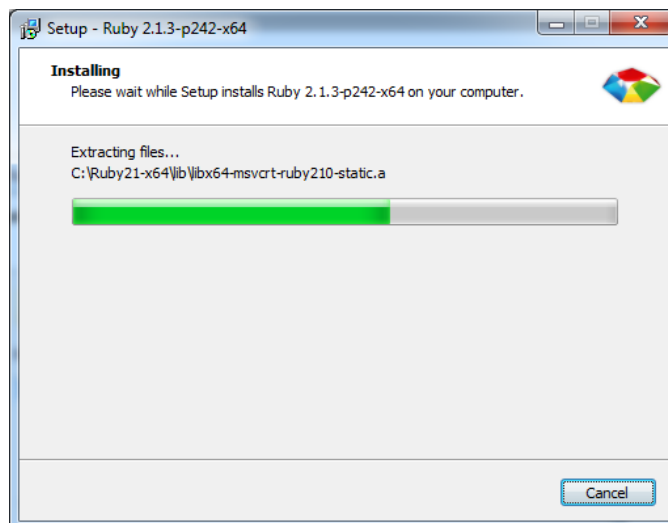
6. Se acepta los términos de licencia



7. Se escoge la opción del medio Add Ruby executables to your PATH



8. Se espera a que el proceso de instalación culmine



9. Por último se da clic en Finish

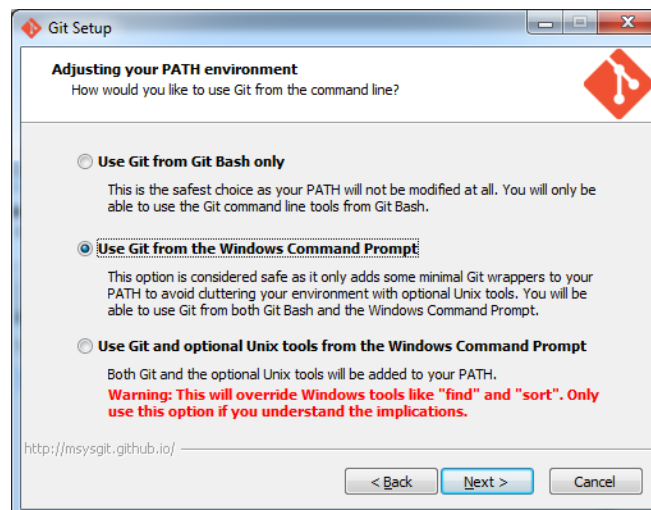


10. Se instala la herramienta Git la que me permitirá trabajar con la plataforma OpenShift.

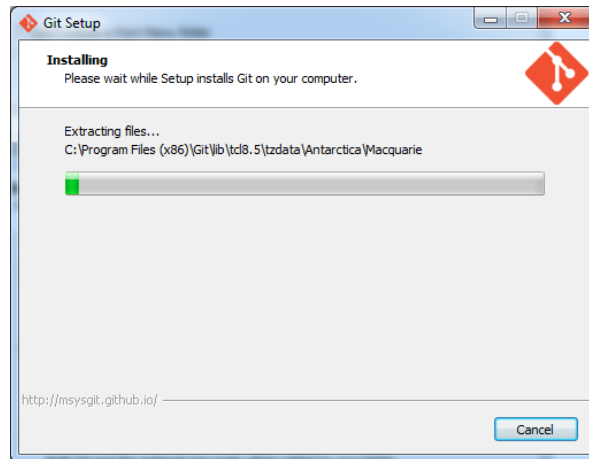


11. Se procede a dar clic en Next en las próximas 2 ventanas dejando las opciones por defecto.

12. Se escoge la opción de Use GIT from the Windows, y clic en Next.



13. Se espera a que el proceso de instalación termine.



14. Se da clic en Finish.



15. Se abre un símbolo del sistema y se verifica si se instaló Rubí con el comando `ruby -e 'puts "Welcome to Ruby"'` Welcome to Ruby

```
C:\Users\Goby>ruby -e 'puts "Welcome to Ruby"' Welcome to Ruby_
```

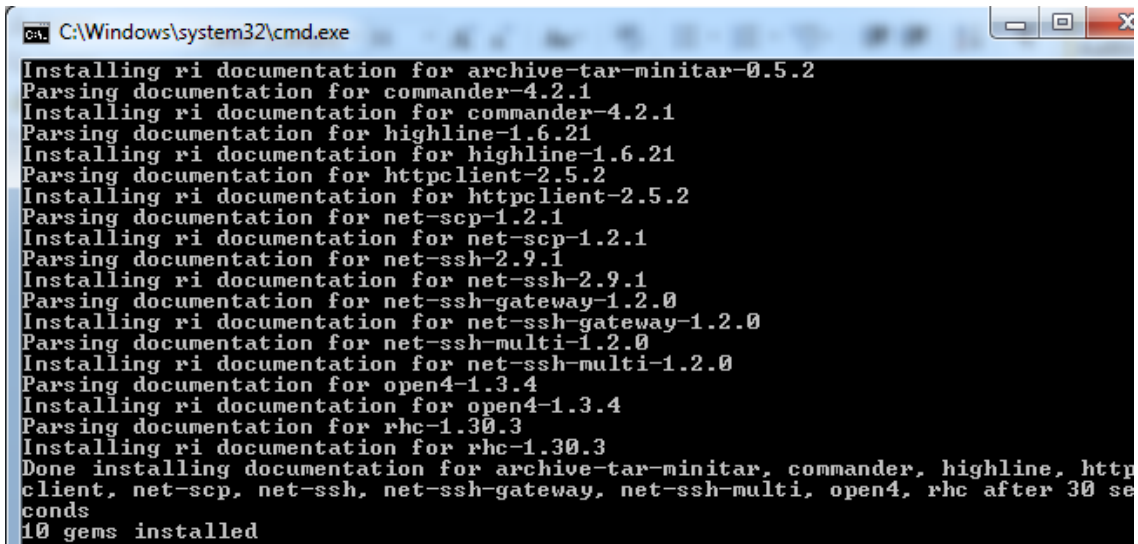
16. Se confirma si se instaló GIT con el comando `git --version` git version 1.7.11.msysgit.1

```
C:\Users\Goby>ruby -e 'puts "Welcome to Ruby"' Welcome to Ruby
Welcome to Ruby
C:\Users\Goby>git --version
git version 1.7.11.msysgit.1
```

17. Se instala GEM con el comando `gem install rhc`

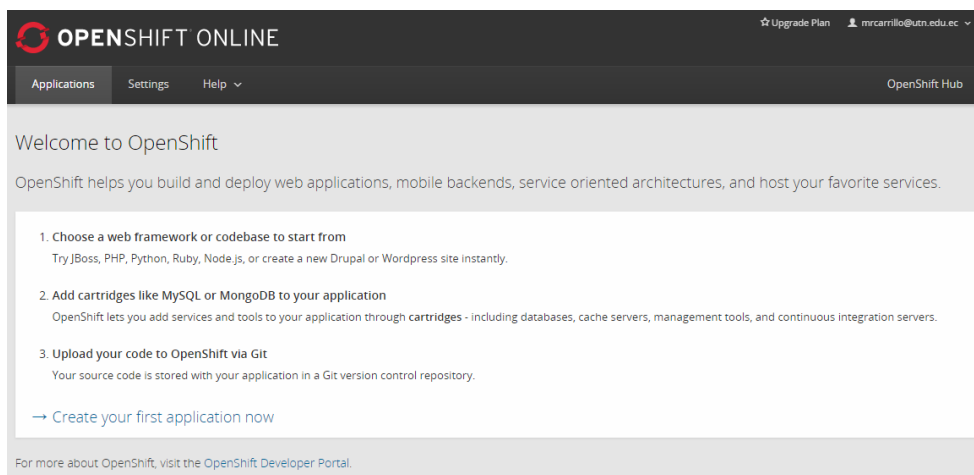
```
C:\Users\Goby>gem install rhc
Fetching: net-ssh-2.9.1.gem < 35%>_
```

18. Se espera que termine de instalarse 10 paquetes.

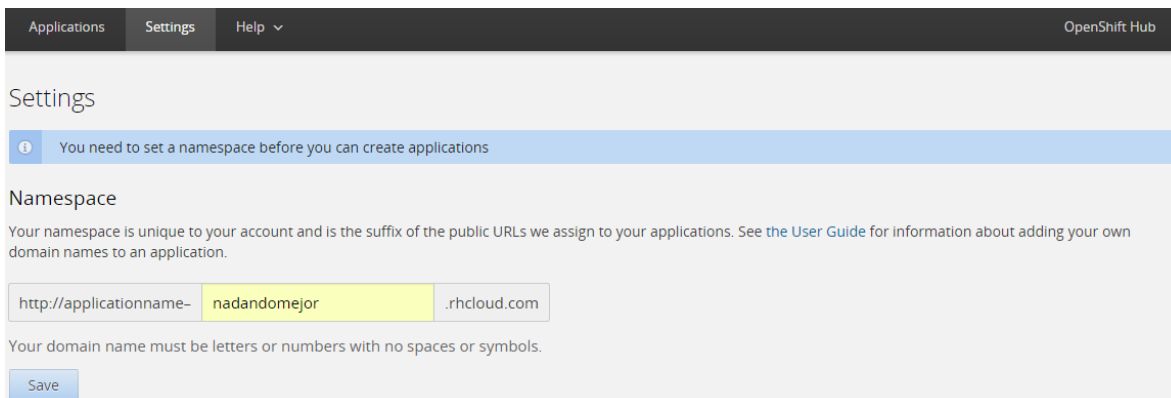


```
ca: C:\Windows\system32\cmd.exe
Installing ri documentation for archive-tar-minitar-0.5.2
Parsing documentation for commander-4.2.1
Installing ri documentation for commander-4.2.1
Parsing documentation for highline-1.6.21
Installing ri documentation for highline-1.6.21
Parsing documentation for httpclient-2.5.2
Installing ri documentation for httpclient-2.5.2
Parsing documentation for net-scp-1.2.1
Installing ri documentation for net-scp-1.2.1
Parsing documentation for net-ssh-2.9.1
Installing ri documentation for net-ssh-2.9.1
Parsing documentation for net-ssh-gateway-1.2.0
Installing ri documentation for net-ssh-gateway-1.2.0
Parsing documentation for net-ssh-multi-1.2.0
Installing ri documentation for net-ssh-multi-1.2.0
Parsing documentation for open4-1.3.4
Installing ri documentation for open4-1.3.4
Parsing documentation for rhc-1.30.3
Installing ri documentation for rhc-1.30.3
Done installing documentation for archive-tar-minitar, commander, highline, http
client, net-scp, net-ssh, net-ssh-gateway, net-ssh-multi, open4, rhc after 30 se
conds
10 gems installed
```

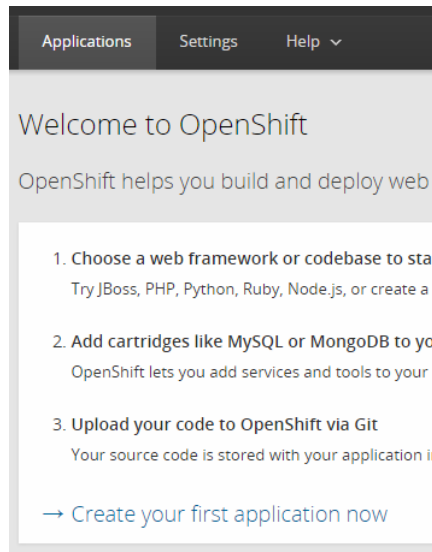
19. Se inicia sesión en OpenShift



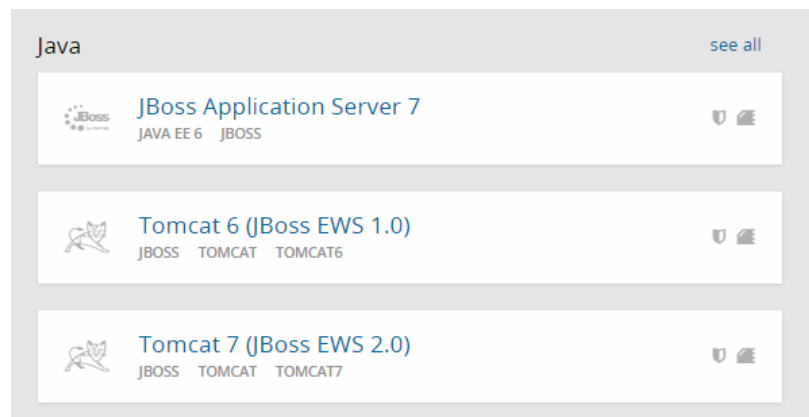
20. Se da clic en Sttings y se escribe un nombre para el dominio, y se da clic en save.



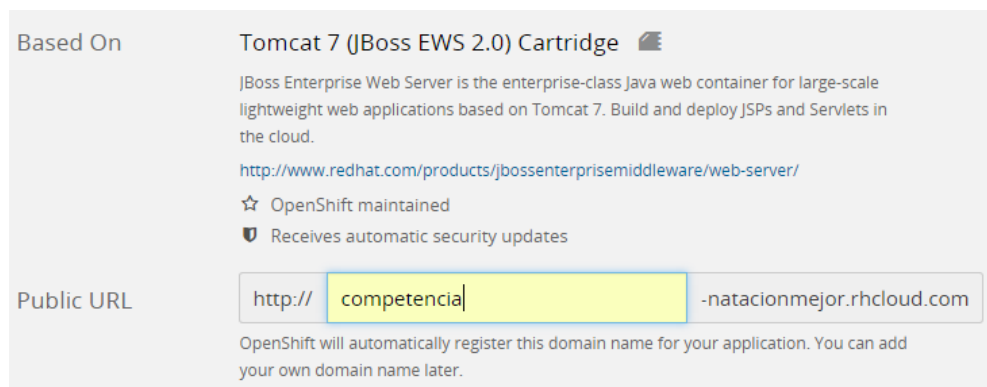
21. Se vuelve a la sección de Applications y se da clic en Create your first application now



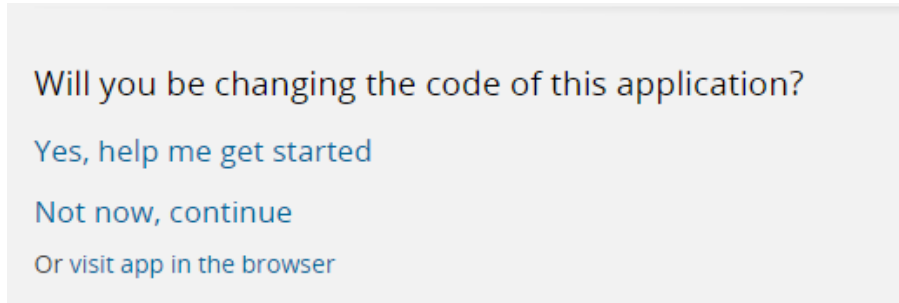
22. Se selecciona un tipo de aplicación y el servicio que usará, en este caso la aplicación es de tipo JAVA y el servidor Apache Tomcat 7



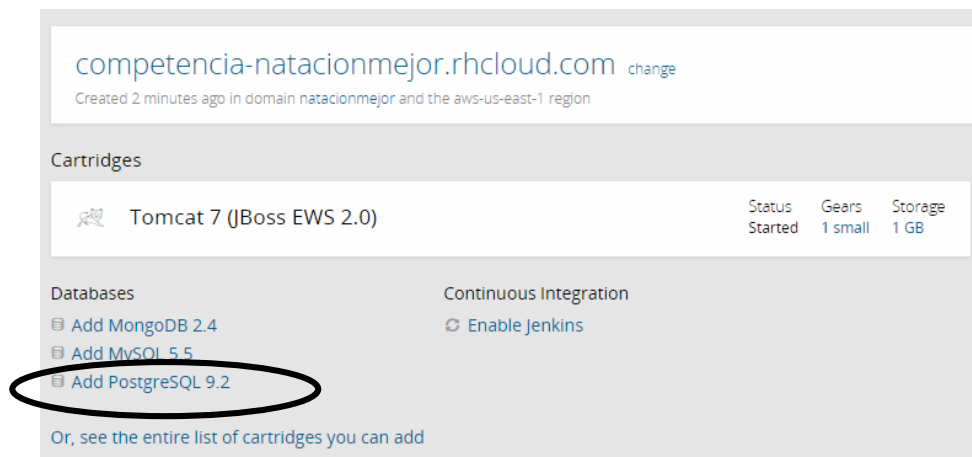
23. Se dirigirá a una ventana donde se dará un nombre de URL pública para el proyecto, seguido del dominio creado en el paso 20, se va hasta el final y se hace clic en Create Application.



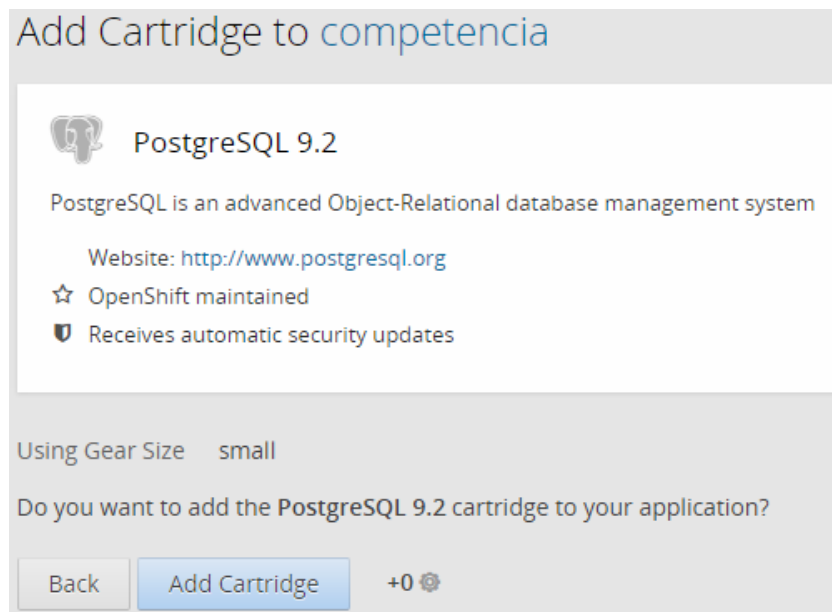
24. Se mostrará el siguiente mensaje donde se pregunta si se desea cambiar el código de la aplicación, se da clic en la opción de la mitad.



25. Se mostrará un resumen de los componentes que tiene la aplicación, en la parte de base de datos se añade PostgreSQL 9.2



26. Se da clic en Add Cartridges.



27. Se mostrará los componentes de la aplicación y los respectivos datos para su acceso

competencia-natacionmejor.rhcloud.com change Started 1 ⚙️ ↻
Created 6 minutes ago in domain natacionmejor and the aws-us-east-1 region

✓ PostgreSQL 9.2 database added. Please make note of these credentials:

Root User: adminybjfqfq
Root Password: 6SRIFGIU3cqd
Database Name: competencia
Connection URL: postgresql://\$OPENSHIFT_POSTGRES_DB_HOST:\$OPENSHIFT_POSTGRES_DB_PORT

Cartridges	Status	Gears	Storage
Tomcat 7 (JBoss EWS 2.0)	Started	1 small	1 GB
PostgreSQL 9.2	Database: competencia	User: adminybjfqfq	Password: show

Source Code
You must add an SSH public key to your account before you can upload code or remotely access your application.

Remote Access
Requires a public key.

Delete this application...

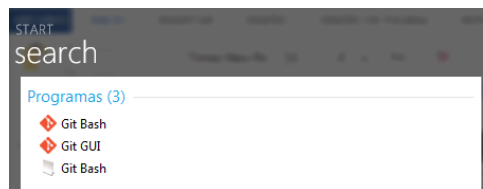
28. En la parte derecha se hace clic en **add an SSH public key to your account** y se da un nombre a nuestra llave

Add a public key

Key name *

competencia

29. Para obtener la llave, se abre el programa Git GUI previamente ya instalado en el sistema



30. Se da clic en help y se selecciona la opción Show SSH Key luego se copia la clave la misma que será ingresada en la página de Openshift.

Git Gui

Repository Help

Create New Repository
Clone Existing Repository
Open Existing Repository

Quit

Your OpenSSH Public Key

Found a public key in: ~/.ssh/id_rsa.pub Generate Key

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDkyJ89w0C7sj66U
kwea/fgyz/hJuSkhmlQ9YvcN4TlnXo4zX0ikuly1XkUkoJNQuyQ1iiljb5
JQ4PZgB5+Xpy3YbybYWC3LKr8O5hKTzk5eDZ62eFwZekT7pyvpQLy
ZoeEryRaISJZ7bJQ7PkFwxGbUu4g6Q3GiWFFRfJa8tvK2xW3WfBhm
qydgxZ8DH+rP5Hk2Jvsmg9wRqnKBN2Wac7fYSikr2OFOZ3NqLV+Np
ypvpL/yCUllzLnYxunJi77bfASUmPPDR2PbS4jSDagpXlZNOphWnyVn
akKjH/t4s/sSxv49FYClbR4cOIC5KGJUyZgM4dK6/65ITLSS7Rti OpenS
hift-Key

Copy To Clipboard Close

31. Se pega y se da clic en create y se mostrará el mensaje de creación.

Add a public key

Key name *

competencia

Paste the contents of your public key file

```
9YvcN4TLnXo4zX0ikuly1XkUkojNQyuQ1iiLjb5JQ4PZgB5+Xpy3YbybYWC3LKr8O5h
KTzk5eDZ62eFwZekT7pyvpQLyZoeERYRaISjZ7bjQ7PkFwGbUu4g6Q3GIWFFRjFja
8tvK2xW3WEfBhmqydgxZ8DH+rP5Hx2jVsmg9wRqnKBN2Wac7fYSikr2OFOZ3Nq
LV+NPypvpL/yCUIlz1nYxunji77bfASUmPPDR2PbS4jSDagpXlznOphWnyVnakKjH/t
4s/sSxV49FYCltbR4cOIC5KGJUYzgM4dK6/65ITLSS7Rtj OpenShift-Key|
```

Cancel Create

Applications Settings Help OpenShift Hub

Settings

✓ Your public key has been created

Public Keys

OpenShift uses a public key to securely encrypt the connection between your local machine and your application and to authorize you to upload code. [Learn more about SSH keys.](#)

Key name	Type	Contents	
competencia	ssh-rsa	AAAB3Nza..TLSS7Rtj	Delete

Add a new key...

Domains

See the [User Guide](#) for information about adding your own domain names to an application.

natacionmejor	1 application	No other members
---------------	---------------	------------------

32. Se vuelve a la ventana de aplicaciones y se hace clic en la que se creó anteriormente.

Applications Settings Help

Applications

1 of 3

Available in domain natacionmejor

competencia 1

Tomcat 7 (JBoss EWS 2.0), PostgreSQL 9.2

competencia-natacionmejor.rhcloud.com Started 1

Created 15 minutes ago in domain natacionmejor and the aws-us-east-1 region

Cartridges	Status	Gears	Storage
Tomcat 7 (JBoss EWS 2.0)	Started	1 small	1 GB
PostgreSQL 9.2			

Database: competencia User: adminybjfqf Password: show

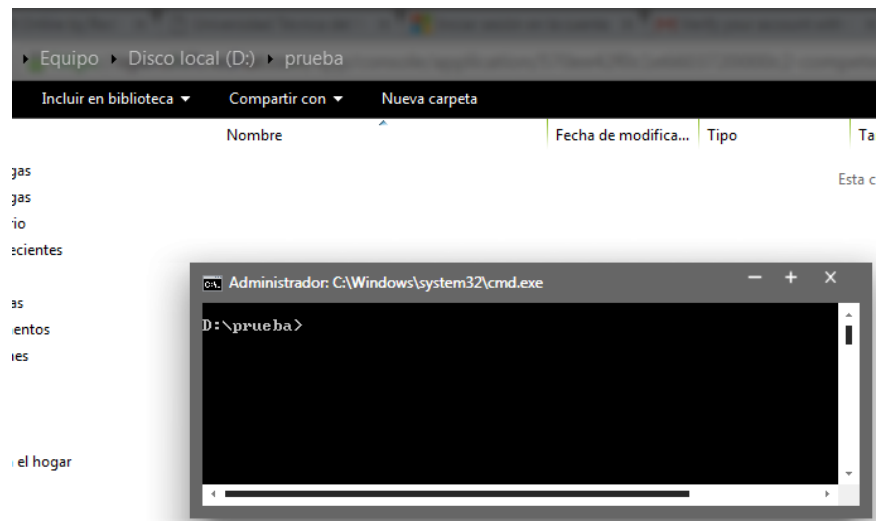
Source Code
ssh://570ee42f0c1e6603720000c2@competencia-nat
Pass this URL to 'git clone' to copy the repository locally.

Remote Access
Want to log in to your application?
Delete this application...

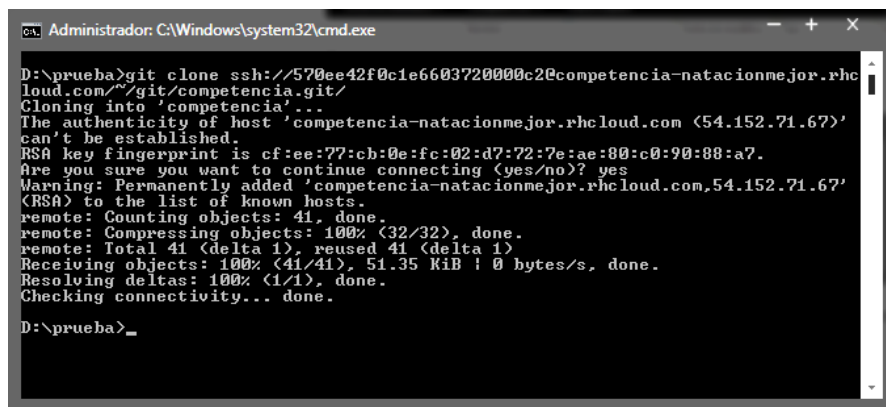
Continuous Integration
Enable Jenkins

Or, see the entire list of cartridges you can add

33. Se procede a crear un clonado de la aplicación de manera local, para poder trabajar de forma más fácil y con el fin de subir la aplicación completa, se ubica en un lugar del equipo y se abre una ventana de comandos.



34. Se escribe el comando git clone seguido de la dirección ssh de la aplicación que se va a clonar (la dirección se encuentra en el lado derecho de la app), y se confirma la conexión.



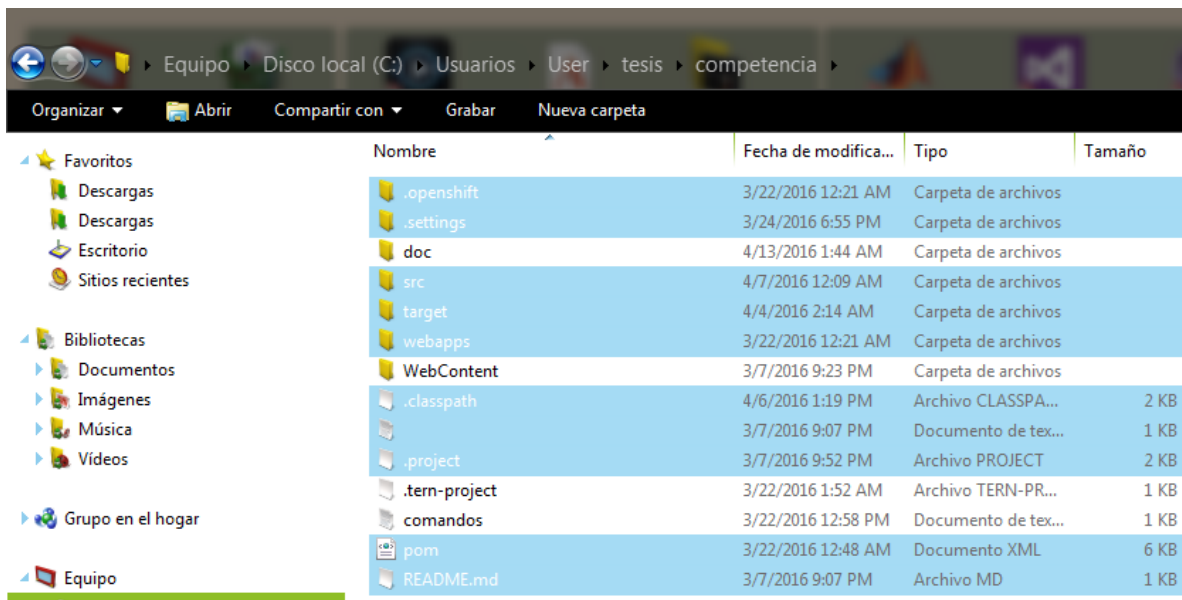
35. Se podrá visualizar que en la carpeta se ha añadido una nueva con el nombre de la aplicación.

Nombre	Fecha de modifica...	Tipo	Tamaño
competencia	4/13/2016 7:50 PM	Carpeta de archivos	

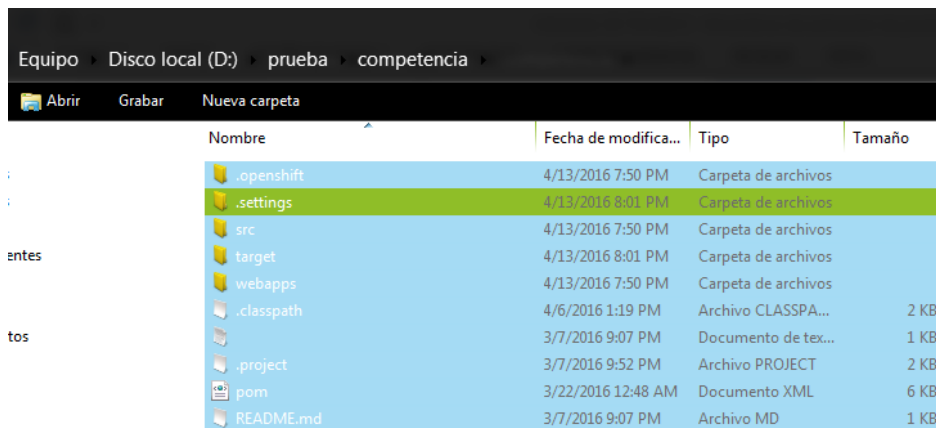
36. Se abre la carpeta y se mostrará la siguiente información

Nombre	Fecha de modifica...	Tipo	Tamaño
.openshift	4/13/2016 7:50 PM	Carpeta de archivos	
src	4/13/2016 7:50 PM	Carpeta de archivos	
webapps	4/13/2016 7:50 PM	Carpeta de archivos	
pom	4/13/2016 7:50 PM	Documento XML	3 KB
README.md	4/13/2016 7:50 PM	Archivo MD	1 KB

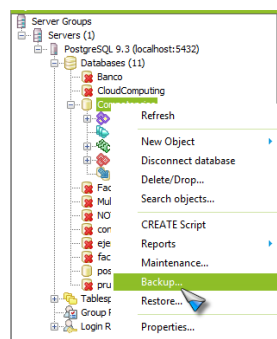
37. Se abre la carpeta del proyecto de eclipse y se copia las siguientes carpetas, en las cuales se encuentra la información referente al sistema como son: las clases Beans, los Manager, las entidades y archivos de configuración.



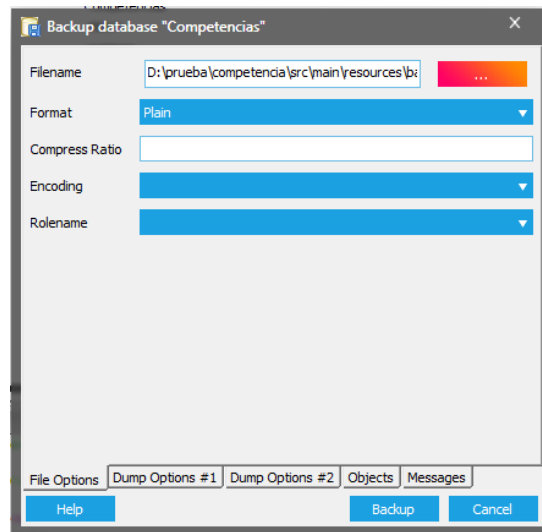
38. Se ubica en la carpeta que se duplicó el sistema y se pega los archivos copiados anteriormente, reemplazando y combinado los mismoa con nombres iguales.



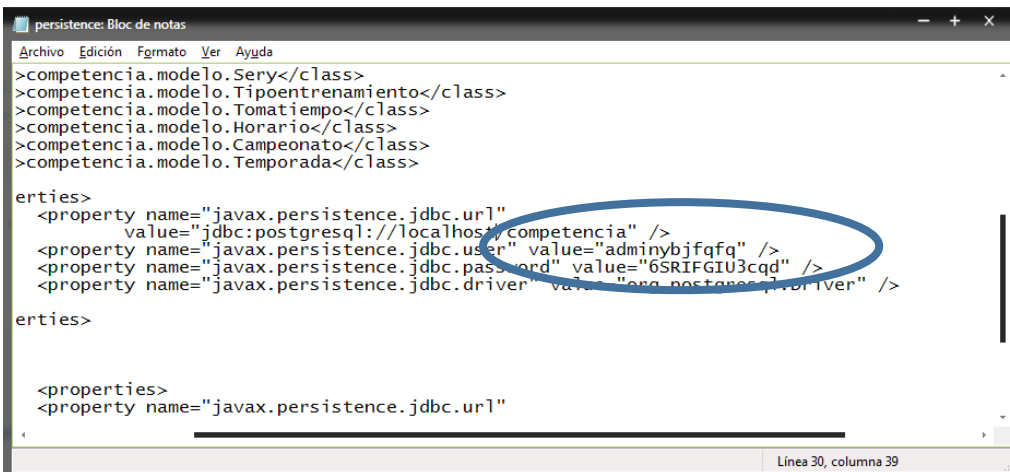
39. Se abre el gestor de base de datos, en este caso Postgres, para crear un backup tipo plano y así poder cargarlo a Openshift, dando clic derecho en la base y se debe dirigir a backup.



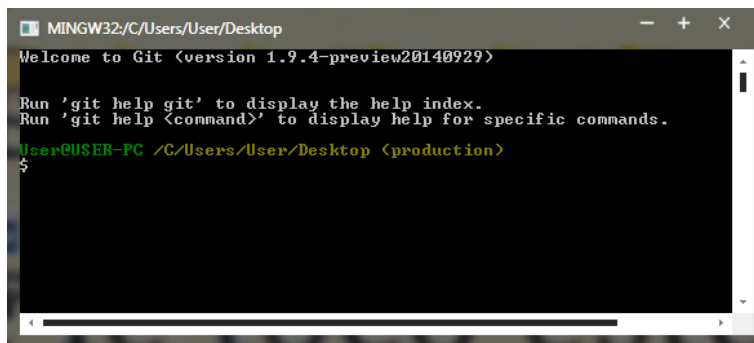
40. En Filename se guarda en cualquier lugar de la carpeta clonada del sistema D:\prueba\competencia\src\main\resources\backup.sql, en format seleccionamos Plain y damos clic en Backup



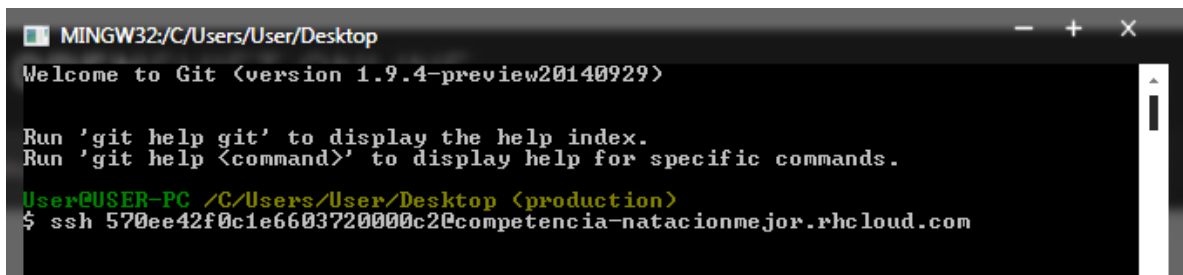
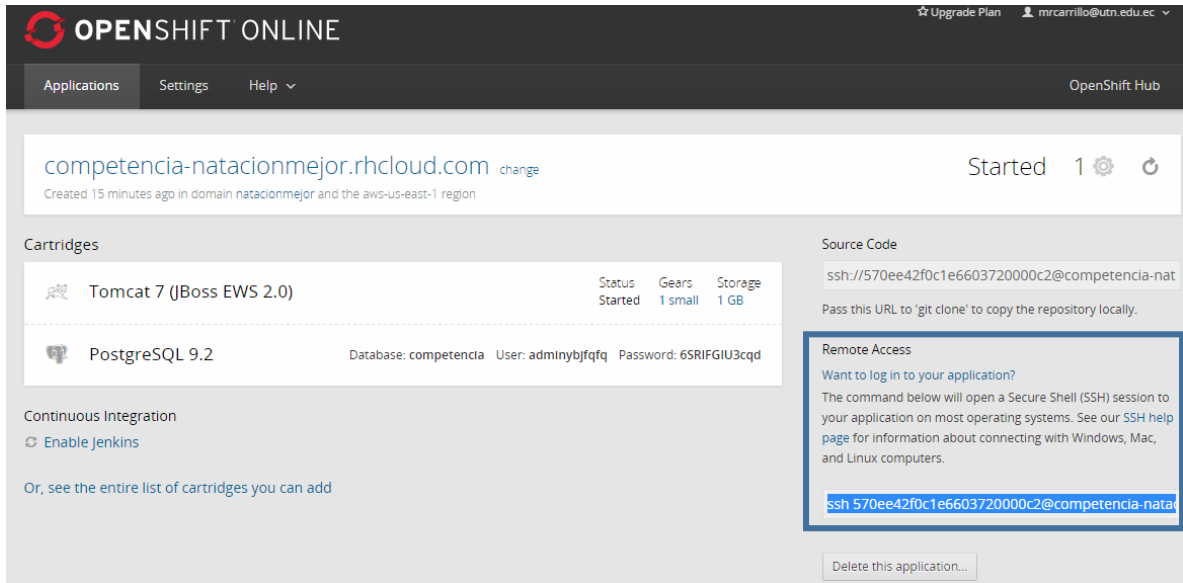
41. Se configura los archivos de persistencia para acceder a la base de datos que se encuentra en Openshift colocando los valores dados en el paso 27.



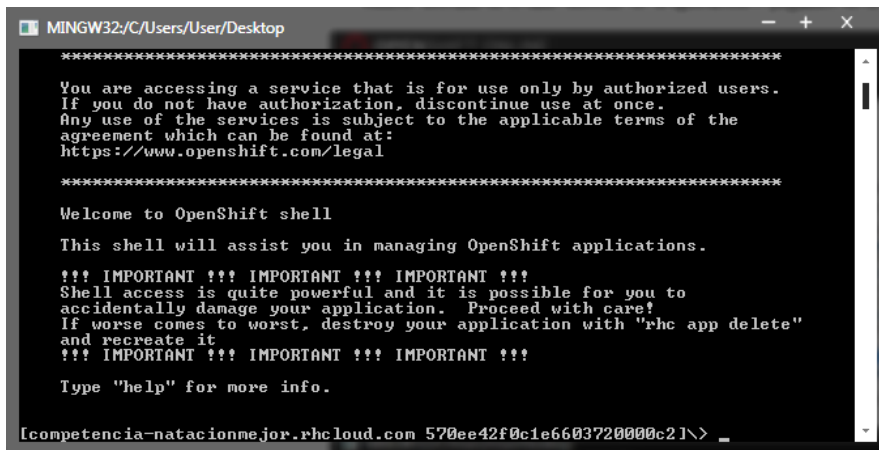
42. En donde dice localhost se cambia por la dirección ip de la base, para obtener la misma se abre el programa Git Bash previamente instalado.



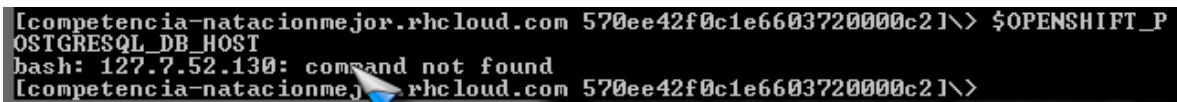
43. Se ingresa a la página de OpenShift en la aplicación y se copia la dirección de acceso remoto ubicada en el lado derecho de la aplicación y se pega en la ventana de Git Bash.



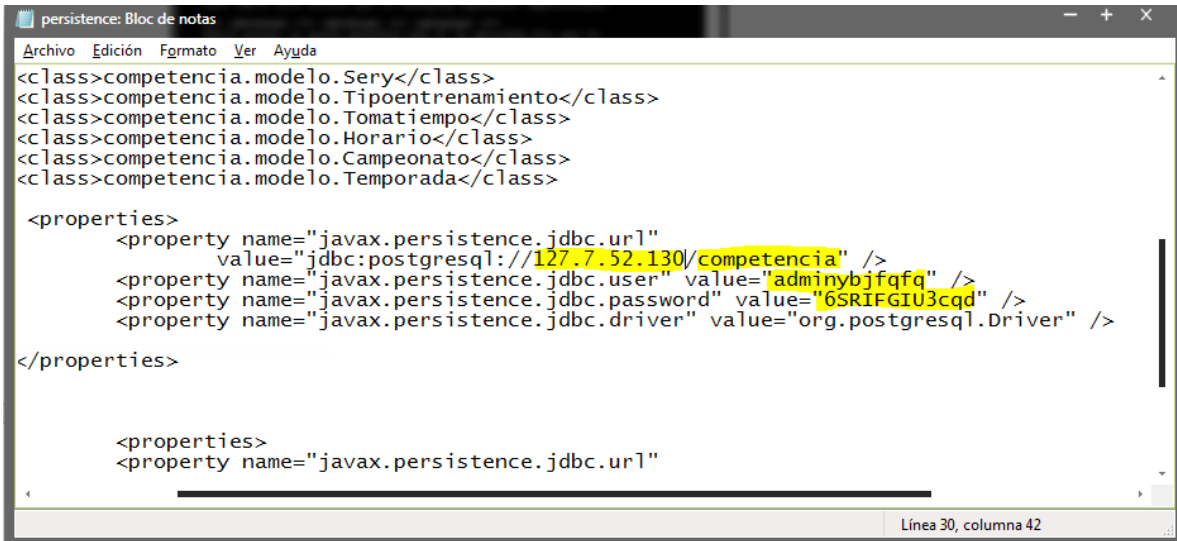
44. Se presiona enter y se espera que establezca conexión:



45. Para obtener la dirección IP de la base de datos, se ingresa el comando \$OPENSHIFT_POSTGRES_DB_HOST y se da enter.



Se mostrará la dirección IP, la misma que se reemplaza en la persistencia por localhost, quedando de la siguiente manera:



```
persistence: Bloc de notas
Archivo Edición Formato Ver Ayuda
<class>competencia.modelo.Sery</class>
<class>competencia.modelo.Tipoentrenamiento</class>
<class>competencia.modelo.Tomatiempo</class>
<class>competencia.modelo.Horario</class>
<class>competencia.modelo.Campeonato</class>
<class>competencia.modelo.Temporada</class>

<properties>
  <property name="javax.persistence.jdbc.url"
    value="jdbc:postgresql://127.7.52.130/competencia" />
  <property name="javax.persistence.jdbc.user" value="adminybjfqq" />
  <property name="javax.persistence.jdbc.password" value="6SRIFGIU3cqd" />
  <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver" />
</properties>

<properties>
  <property name="javax.persistence.jdbc.url"

```

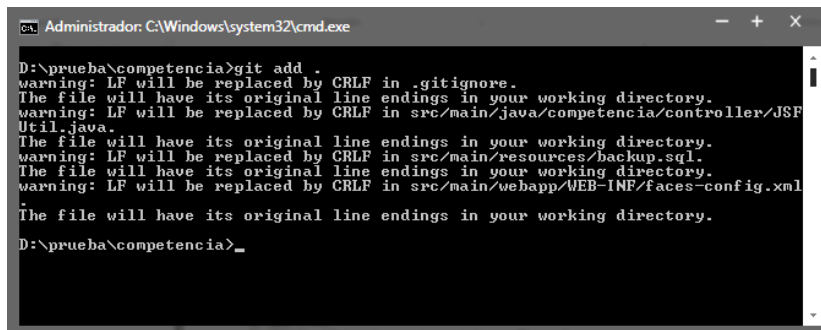
IMPORTANTE: Se debe realizar el cambio en todos los archivos de persistencia.

46. Una vez ejecutado estos cambios, se ubica a la ventana del paso 38 y se abre una ventana de comandos en esa posición.



```
Administrador: C:\Windows\system32\cmd.exe
D:\prueba\competencia>
```

Se ingresa el comando **git add .**



```
Administrador: C:\Windows\system32\cmd.exe
D:\prueba\competencia>git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in src/main/java/competencia/controller/JSF
Util.java.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in src/main/resources/backup.sql.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in src/main/webapp/WEB-INF/faces-config.xml
The file will have its original line endings in your working directory.
D:\prueba\competencia>_
```

Después se digita el comando **git commit -m prueba** la última palabra puede variar en solo una referencia para saber los cambios que se va a subir.

```
Administrador: C:\Windows\system32\cmd.exe
D:\prueba\competencia>git commit -m prueba
[master warning: LF will be replaced by CRLF in src/main/webapp/WEB-INF/faces-co
nfig.xml.
The file will have its original line endings in your working directory.
5c18ccc1 prueba
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in src/main/java/competencia/controller/JSF
Util.java.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in src/main/resources/backup.sql.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in src/main/webapp/WEB-INF/faces-config.xml
.
The file will have its original line endings in your working directory.
142 files changed, 12451 insertions(+), 16 deletions(-)
create mode 100644 .classpath
create mode 100644 .gitignore
create mode 100644 .project
create mode 100644 .settings/.jsdtscope
create mode 100644 .settings/org.eclipse.core.resources.prefs
create mode 100644 .settings/org.eclipse.jdt.core.prefs
create mode 100644 .settings/org.eclipse.jpt.core.prefs
create mode 100644 .settings/org.eclipse.jpt.entitygen.competencia.public
create mode 100644 .settings/org.eclipse.jpt.entitygen.competencias.public
create mode 100644 .settings/org.eclipse.m2e.core.prefs
create mode 100644 .settings/org.eclipse.wst.common.component
create mode 100644 .settings/org.eclipse.wst.common.project.facet.core.prefs.xml
create mode 100644 .settings/org.eclipse.wst.common.project.facet.core.xml
create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.container
create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.name
create mode 100644 .settings/org.eclipse.wst.validation.prefs
create mode 100644 .settings/org.hibernate.eclipse.console.prefs
create mode 100644 src/main/java/META-INF/persistence.xml
create mode 100644 src/main/java/competencia/controller/BeanAvances.java
create mode 100644 src/main/java/competencia/controller/BeanCampeonato.java
create mode 100644 src/main/java/competencia/controller/BeanClub.java
create mode 100644 src/main/java/competencia/controller/BeanDeportista.java
create mode 100644 src/main/java/competencia/controller/BeanEntrenador.java
create mode 100644 src/main/java/competencia/controller/BeanEntrenamiento.java
create mode 100644 src/main/java/competencia/controller/BeanInscripcion.java
create mode 100644 src/main/java/competencia/controller/BeanLogin.java
create mode 100644 src/main/java/competencia/controller/BeanNadadore.java
```

Por último se digita la línea de comando **git push** para publicar.

```
D:\prueba\competencia>git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 169, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (154/154), done.
Writing objects: 98% (158/161), 3.35 MiB | 61.00 KiB/s
```

Se espera que se descarguen todos los paquetes hasta que se muestre el mensaje de construcción exitosa.


```

remote: Downloaded: http://mirror.ops.rhcloud.com/nexus/content/groups/public/or
g/apache/maven/shared/maven-filtering/1.0-beta-2/maven-filtering-1.0-beta-2.jar
(33 KB at 356.1 KB/sec)
remote: [INFO] Packaging webapp
remote: [INFO] Assembling webapp [competencia] in [/var/lib/openshift/570ee42f0c
1e6603720000c2/app-root/runtime/repo/target/competencia]
remote: [INFO] Processing war project
remote: [INFO] Copying webapp resources [/var/lib/openshift/570ee42f0c1e66037200
00c2/app-root/runtime/repo/src/main/webapp]
remote: [INFO] Webapp assembled in [682 msecs]
remote: [INFO] Building war: /var/lib/openshift/570ee42f0c1e6603720000c2/app-roo
t/runtime/repo/webapps/ROOT.war
remote: [INFO] WEB-INF/web.xml already added, skipping
remote: [INFO] -----
remote: [INFO] BUILD SUCCESS
remote: [INFO] -----
remote: [INFO] Total time: 2:10.627s
remote: [INFO] Finished at: Wed Apr 13 21:44:12 EDT 2016
remote: [INFO] Final Memory: 17M/162M
remote: [INFO] -----
remote: Preparing build for deployment
remote: Deployment id is 35fb404d
remote: Activating deployment
remote: Starting Postgres cartridge
remote: Postgres started
remote: Starting jbossews cartridge
remote: Found 127.7.52.129:8080 listening port
remote: -----
remote: Git Post-Receive Result: success
remote: Activation status: success
remote: Deployment completed with status: success
To ssh://570ee42f0c1e6603720000c2@competencia-natacionmejor.rhcloud.com/~/.git/co
mpetencia.git/
6417469..5c18ccc master -> master

```

47. Una vez cargado el sistema a la plataforma se procede a ejecutar el Backup de la base de datos, para ello se abre el programa Git Bash y se ubica en la carpeta que se almacenó el backup.sql.

```

[competencia-natacionmejor.rhcloud.com 570ee42f0c1e6603720000c2]~\> dir
app-deployments app-root git jbossews postgresql
[competencia-natacionmejor.rhcloud.com 570ee42f0c1e6603720000c2]~\> cd app-root/
repo/src/main/resources/
[competencia-natacionmejor.rhcloud.com resources]~\> dir
backup.backup backup.sql META-INF
[competencia-natacionmejor.rhcloud.com resources]~\>

```

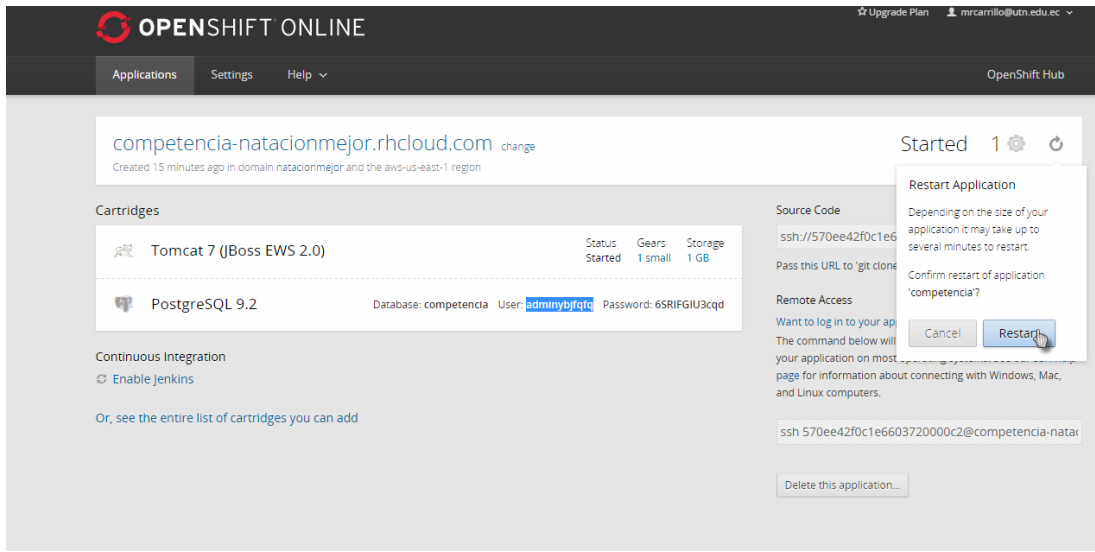
Para cargar el backup se ejecuta la siguiente línea de comando `psql competencia < backup.sql`

```

SET
SET
SET
SET
CREATE EXTENSION
COMMENT
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE

```

48. Por último se regresa a la aplicación en Openshift, y se reinicia la app.



The screenshot shows the OpenShift Online interface. At the top, there's a navigation bar with 'Applications', 'Settings', and 'Help'. The main content area displays the application 'competencia-natacionmejor.rhcloud.com', which was created 15 minutes ago. Below this, there's a 'Cartridges' section with a table:

Cartridge	Status	Gears	Storage
Tomcat 7 (JBoss EWS 2.0)	Started	1 small	1 GB
PostgreSQL 9.2			

Additional details for PostgreSQL 9.2: Database: competencia, User: adminybjfqt, Password: 6SRIFGIU3cqd. There are also sections for 'Continuous Integration' (with 'Enable Jenkins' checked) and 'Remote Access' (with an SSH command: `ssh 570ee42f0c1e660372000c2@competencia-nata...`). A 'Restart Application' dialog box is overlaid on the right, asking for confirmation to restart the application.

Se da clic en el enlace de la aplicación y se podrá acceder al sistema.

