

Actividad Final Curso Fundamentos

Instructor

Ingeniero Cesar Hidalgo

Estudiante

Juan David Henao Estrada

Semillero JAVA

Bogotá D.C

2022

Introducción

El Scrum es una metodología ágil y flexible que se utiliza para gestionar proyectos, su principal objetivo es apoyar proyectos donde el entorno de estos es muy cambiante en cuanto a tecnologías, requerimientos y equipos.

En lo posterior se realizó una definición de roles, el product backlog, el sprint planning y entregable. Detallando aspectos de las funciones en el proceso a realizar.

Definición de los roles y el equipo Scrum

Roles y Equipo					
Nombre	Conocimientos	Cargo	Disponible	Rol	Descripción
Cesar Hidalgo	Conocimiento en la lógica de negocio	Representante ante de dueño del negocio (gimnasio)	Miércoles de 5pm a 6pm	Product Owner	Es el dueño del producto su tarea principal es la de maximizar el valor del producto resultante
Juan David Henao	Master Scrum	Enlace entre product owner y el equipo de desarrollo	Tiempo completo	Scrum Máster	Es el líder SCRUM TEAM y de la organización en general.
Juan David Henao	Desarrollo y análisis	Back End y Front End	Tiempo completo	Developer	Responsable del desarrollo del proyecto.

Product Backlog

PRODUCT BACKLOG					
Sprints	PBI	Historia	Criterio de Aceptación	puntaje	prioridad
Sprint 1	SUX-1	Como product owner quiero un software que permita hacer inicio de sesion.	El ingreso será mediante credenciales validas para seguridad de las pesonas usan el sistema	4	10
	SUX-2	Como product owner quiero un software que permita hacer registro de personas según su rol.	El sistema solo permite interactuar si se está registrado.	5	10
Sprint 2	SUX-3	Como usuario quiero registrar mi informacion ademas datos de contacto de emergencia y mis records de peso.	El usuario podrá ingresar sus propios datos en el formulario como tambien records de peso	5	8
	SUX-4	Como coach debo poder ver quien agenda clase y ajustar el contenido de las clases que se dictan	El coach visualizará agenda de clases y ajustar contenido	3	9
	SUX-5	Como administrador debo ver quien tiene activa la membresia y la informacion de los usuarios.	El administrador será el unico que puede tener acceso a la informacion de usuarios y estado de la membresia	4	10
	SUX-6	Como product owner quiero un software que me permita reservar clases de un gimnasio	El administrador y coach podrán reservar clases al usuario o el mismo usuario tambien podrá despues de estar registrado e iniciado sesion	6	10

PBI (Product Backlog Item)

SUX-x (Storie User eXperience - x (reemplace la x por el numero))

Sprint Planning

En el orden de ideas, para lograr los objetivos se inicia trabajando en la tabla Sprint Backlog en el Sprint 1, allí en el PBI encontramos dos SUX-x. Comenzamos por el PBI SUX-1 inicio de sesión y luego continuamos con el PBI SUX-2 formularios de registro personas según rol, con una duración de 5 días en total.

Después de finalizado el Sprint1 de la tabla Sprint Backlog continuamos con la tabla Sprint Backlog pero Sprint2, de no lograr terminar algún SUX-x del Sprint1, se deberá priorizar ese SUX-x y se colocaría en primer lugar en la tabla Sprint Backlog Sprint2, de haber finalizado completamente el Sprint1 del Sprint Backlog se continua con el Sprint2 en su orden normal.

Ver tabla Sprint Backlog a continuación:

Sprint Backlog

SPRINT	INICIO	DURACION
Sprint 2	7/11/2022	5 días

SPRINT BACKLOG						
PBI	Actividad	Responsable	Dificultad	Fecha Entrega	NO Completada	Estado
SUX-3	Modificacion o actualizacion formulario registro de datos que requiere el usuario con sus atributos	Juan David H	Media	8/11/2022	Ninguna	Completada
SUX-4	Creacion visual espacio coach	Juan David H	Alta	9/11/2022	Ninguna	Completada
SUX-5	Ceacion visual espacio administrador para ver informacion de usuarios y estado membresia	Juan David H	Alta	10/11/2022	Ninguna	En proceso
SUX-6	Creacion visual espacio o interfaz para la reserva de clases	Juan David H	Alta	10/11/2022	Ninguna	En proceso

NO Completada : en caso de que en el Sprint1 NO se haya completado alguna actividad debe colocarse de primera

Entregable

Con el fin de no desperdiciar recursos y dar claridad en lo que sería el producto mínimo entregable, podríamos decir que cuenta con un diseño intuitivo de fácil uso para cualquier usuario sin experiencia en el manejo de computadores.

Este producto cuenta con las funciones necesarias para solucionar un problema específico.

Inicio de sesión:

BIENVENIDOS

Gimnasio GYM

Inicio Sesión

Usuario:

 Ingrese usuario

Contraseña:

 Ingrese Clave

Ingresar

Diseñado y Desarrollado por Juan David Henao « Copyright © 2022 » • Rionegro-Antioquia.


Registro usuario:


GIMNASIO GYM


Registrar Usuario


[Volver](#)


Ingresar Datos del Usuario


 Ingrese numero de identificación


 Ingrese nombres

 Ingrese apellidos

 Ingrese dirección domicilio

 Ingrese teléfono emergencia

 Ingrese record peso

 Mínimo 5 caracteres para clave

Registrar Usuario

Diseñado y Desarrollado por Juan David Henao « Copyright © 2022 » • Rionegro-Antioquia.


Registro coach:


GIMNASIO GYM


Registrar Coach


[Volver](#)


Ingresar Datos del Coach


 Ingrese numero de identificación


 Ingrese nombres

 Ingrese apellidos

 Ingrese dirección domicilio

 Ingrese teléfono

 Ingrese correo

 Mínimo 5 caracteres para clave

Registrar Coach

Diseñado y Desarrollado por Juan David Henao « Copyright © 2022 » • Rionegro-Antioquia.


Registro Administrador:


GIMNASIO GYM

Registrar Administrador


[Volver](#)

Ingresar Datos del Administrador


 Ingrese numero de identificación




Ingrese nombres




Ingrese apellidos




Ingrese dirección domicilio



Ingrese teléfono



Ingrese correo



Mínimo 5 caracteres para clave

Registrar Administrador

Diseñado y Desarrollado por Juan David Henao « Copyright © 2022 » • Rionegro-Antioquia.


Registro clases:


GIMNASIO GYM

Registrar Clases


[Volver](#)

Ingresar Datos para Registrar la Clase


 Ingrese nombre de la clase




Ingrese Día



Ingrese Hora



Ingrese Coach

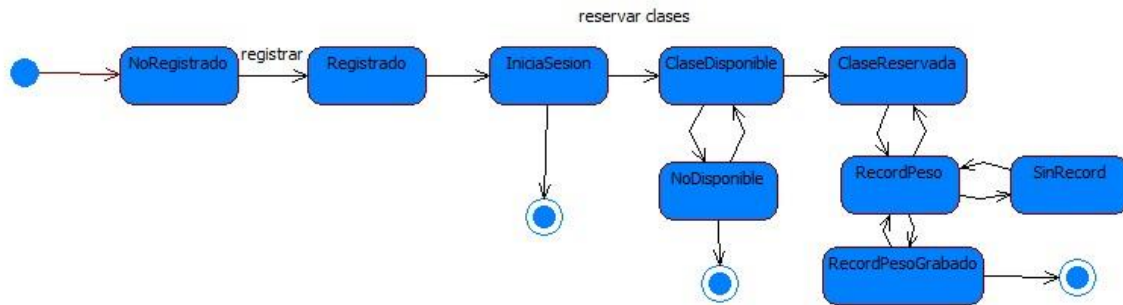


Ingrese nombre del Alumno

Registrar Clase

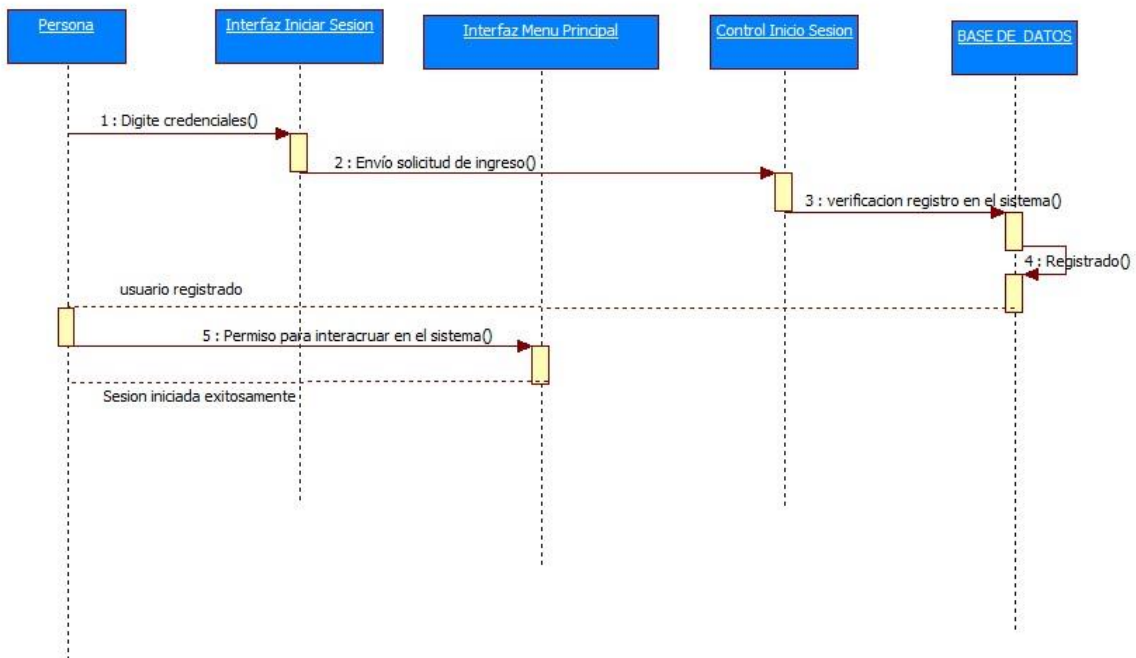
Diseñado y Desarrollado por Juan David Henao « Copyright © 2022 » • Rionegro-Antioquia.

Diagramas de estado

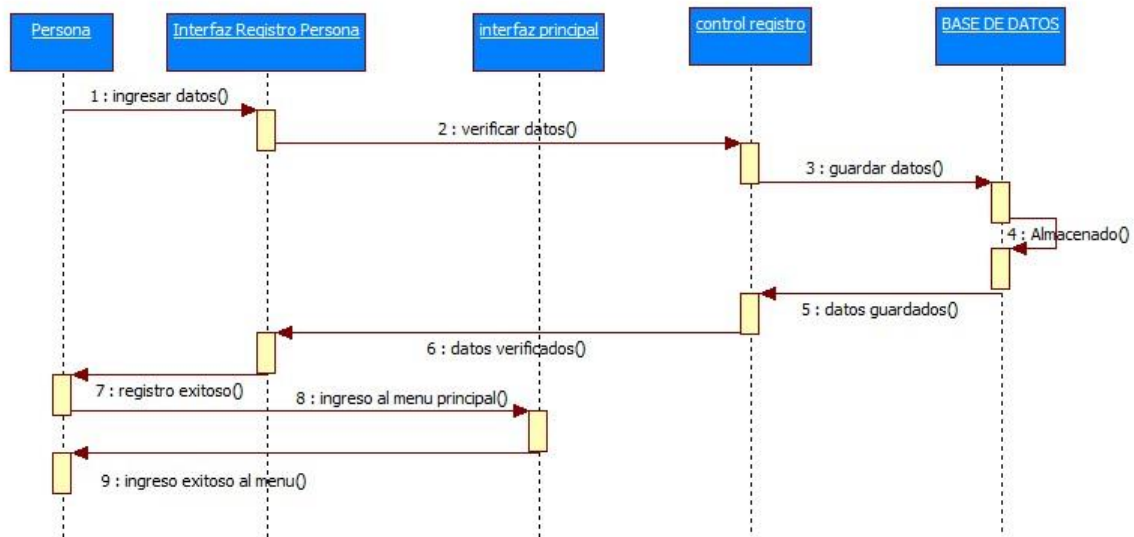


Diagramas de secuencia:

Secuencia Iniciar sesión



Secuencia Registro persona



Secuencia Registro Clases

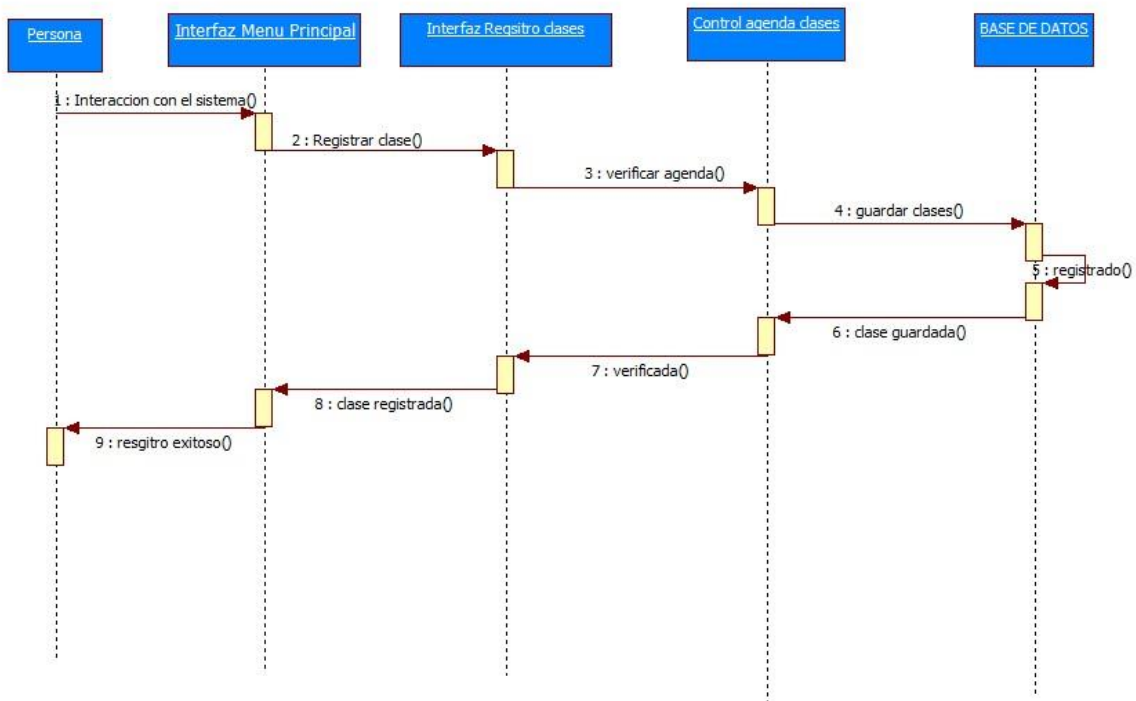


Diagrama clases

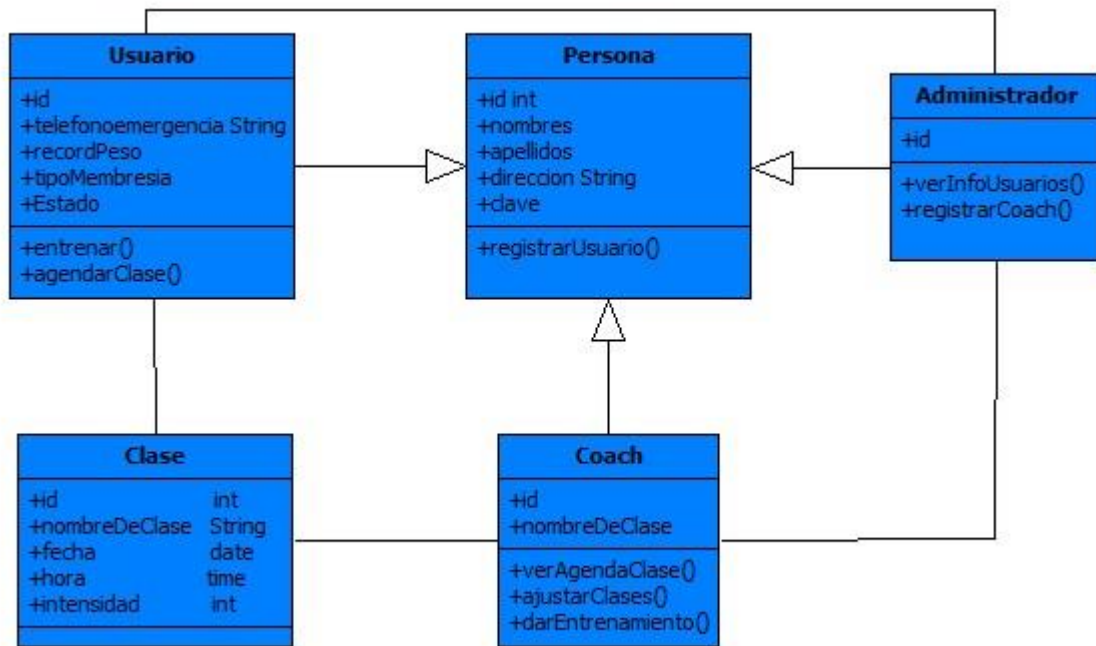
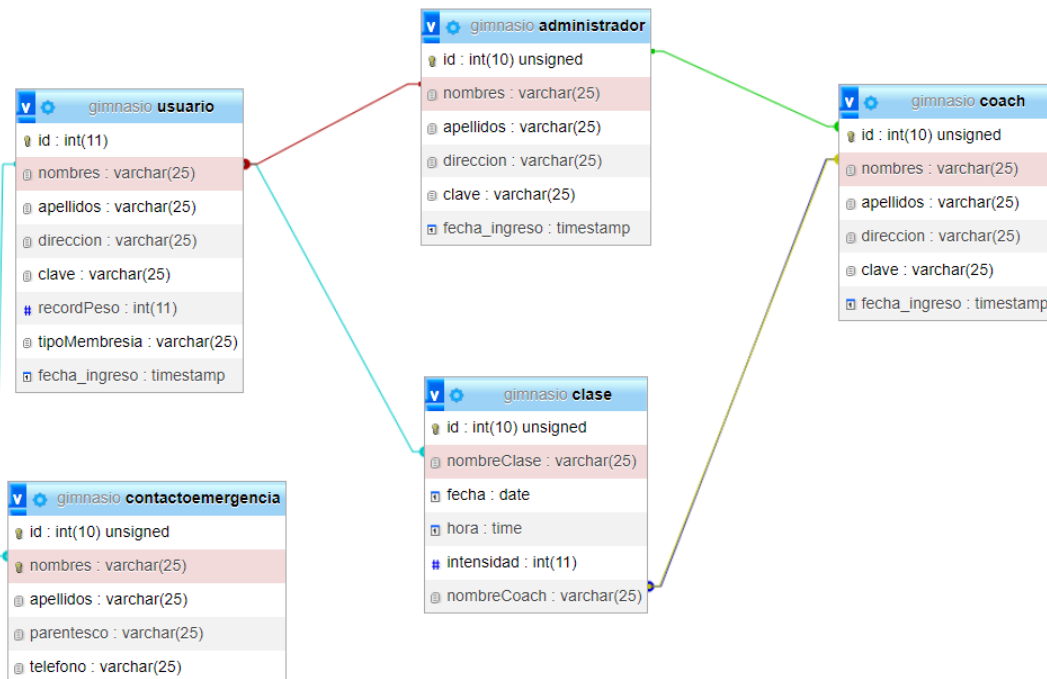


Diagrama entidad relación



Crear_las_Tablas

-- Base de datos: gimnasio

--

CREATE DATABASE IF NOT EXISTS gimnasio DEFAULT CHARACTER SET
utf8mb4 COLLATE utf8mb4_general_ci;

USE gimnasio;

-- Estructura de tabla para la tabla administrador

--

CREATE TABLE administrador (
id int(10) UNSIGNED NOT NULL,
nombres varchar(25) NOT NULL,
apellidos varchar(25) NOT NULL,
direccion varchar(25) NOT NULL,
clave varchar(25) NOT NULL,
fecha_ingreso timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Estructura de tabla para la tabla clase

CREATE TABLE clase (
id int(10) UNSIGNED NOT NULL,
nombreClase varchar(25) NOT NULL,

fecha date NOT NULL,
hora time NOT NULL,
intensidad int(11) NOT NULL,
nombreCoach varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Estructura de tabla para la tabla coach

--

CREATE TABLE coach (
id int(10) UNSIGNED NOT NULL,
nombres varchar(25) NOT NULL,
apellidos varchar(25) NOT NULL,
direccion varchar(25) NOT NULL,
clave varchar(25) NOT NULL,
fecha_ingreso timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Estructura de tabla para la tabla contactoemergencia

--

CREATE TABLE contactoemergencia (
id int(10) UNSIGNED NOT NULL,
nombres varchar(25) NOT NULL,
apellidos varchar(25) NOT NULL,

_parentesco varchar(25) NOT NULL,

_telefono varchar(25) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Estructura de tabla para la tabla usuario

==

CREATE TABLE usuario (

_id int(11) NOT NULL,

_nombres varchar(25) NOT NULL,

_apellidos varchar(25) NOT NULL,

_direccion varchar(25) NOT NULL,

_clave varchar(25) NOT NULL,

_recordPeso int(11) NOT NULL,

_tipoMembresia varchar(25) NOT NULL,

_fecha_ingreso timestamp NOT NULL DEFAULT current_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

==

-- Índices para tablas volcadas

==

==

-- Indices de la tabla administrador

==

ALTER TABLE administrador

_ADD PRIMARY KEY (id),

_ADD KEY nombres (nombres);

--

-- Indices de la tabla clase

--

ALTER TABLE clase

_ADD PRIMARY KEY (id),

_ADD KEY nombreCoach (nombreCoach),

_ADD KEY nombreClase (nombreClase);

--

-- Indices de la tabla coach

--

ALTER TABLE coach

_ADD PRIMARY KEY (id),

_ADD KEY nombres (nombres);

--

-- Indices de la tabla contactoemergencia

--

ALTER TABLE contactoemergencia

_ADD PRIMARY KEY (id),

_ADD UNIQUE KEY nombres (nombres);

--

-- Indices de la tabla usuario

--

ALTER TABLE usuario

ADD PRIMARY KEY (id),

ADD KEY nombres (nombres);

--

-- AUTO INCREMENT de las tablas volcadas

--

--

-- AUTO INCREMENT de la tabla administrador

--

ALTER TABLE administrador

MODIFY id int(10) UNSIGNED NOT NULL AUTO INCREMENT;

--

-- AUTO INCREMENT de la tabla clase

--

ALTER TABLE clase

MODIFY id int(10) UNSIGNED NOT NULL AUTO INCREMENT;

--

-- AUTO INCREMENT de la tabla coach

==

ALTER TABLE coach

MODIFY id int(10) UNSIGNED NOT NULL AUTO INCREMENT;

==

-- AUTO INCREMENT de la tabla contactoemergencia

==

ALTER TABLE contactoemergencia

MODIFY id int(10) UNSIGNED NOT NULL AUTO INCREMENT;

==

-- AUTO INCREMENT de la tabla usuario

==

ALTER TABLE usuario

MODIFY id int(11) NOT NULL AUTO INCREMENT;

==

-- Restricciones para tablas volcadas

==

==

-- Filtros para la tabla clase

==

ALTER TABLE clase

ADD CONSTRAINT clase_ibfk_1 FOREIGN KEY (nombreCoach) REFERENCES coach (nombres) ON UPDATE CASCADE,

ADD CONSTRAINT clase_ibfk_2 FOREIGN KEY (nombreClase) REFERENCES usuario (nombres) ON UPDATE CASCADE;

==

-- Filtros para la tabla coach

==

ALTER TABLE coach

ADD CONSTRAINT coach_ibfk_1 FOREIGN KEY (id) REFERENCES administrador (id) ON DELETE NO ACTION ON UPDATE NO ACTION,

ADD CONSTRAINT coach_ibfk_2 FOREIGN KEY (nombres) REFERENCES clase (nombreCoach) ON UPDATE CASCADE;

==

-- Filtros para la tabla contactoemergencia

==

ALTER TABLE contactoemergencia

ADD CONSTRAINT contactoemergencia_ibfk_1 FOREIGN KEY (nombres) REFERENCES usuario (nombres) ON UPDATE CASCADE;

-- Filtros para la tabla usuario

==

ALTER TABLE usuario

ADD CONSTRAINT usuario_ibfk_1 FOREIGN KEY (nombres) REFERENCES administrador (nombres) ON UPDATE CASCADE;

SET FOREIGN KEY CHECKS=1;

Crear las Clases

-Clase padre Persona

```
package co.com.nttdata.gimnasio;

public class Persona {

    private int id;
    private String nombres;
    private String apellidos;
    private String direccion;
    private String clave;

    public Persona(int id, String nombres, String apellidos, String
direccion, String clave) {
        this.id = id;
        this.nombres = nombres;
        this.apellidos = apellidos;
        this.direccion = direccion;
        this.clave = clave;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNombres() {
        return nombres;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public String getDireccion() {
        return direccion;
    }

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

```

    public String getClave() {
        return clave;
    }

    public void setClave(String clave) {
        this.clave = clave;
    }

    public void registrarUsuario(){

    }
}

```

Clase hija Usuario:

```

package co.com.nttdata.gimnasio;

public class Usuario extends Persona{

    private int idu;
    private String telefonoEmergencia;
    private String recordPeso;
    private String tipoMembresia;
    private String estado;

    public Usuario(int id, String nombres, String apellidos, String
direccion, String clave, int idu, String telefonoEmergencia, String
recordPeso, String tipoMembresia, String estado) {
        super(id, nombres, apellidos, direccion, clave);
        this.idu = idu;
        this.telefonoEmergencia = telefonoEmergencia;
        this.recordPeso = recordPeso;
        this.tipoMembresia = tipoMembresia;
        this.estado = estado;
    }

    public int getIdu() {
        return idu;
    }

    public void setIdu(int idu) {
        this.idu = idu;
    }

    public String getTelefonoEmergencia() {
        return telefonoEmergencia;
    }

    public void setTelefonoEmergencia(String telefonoEmergencia) {
        this.telefonoEmergencia = telefonoEmergencia;
    }

    public String getRecordPeso() {
        return recordPeso;
    }
}

```

```

    }

    public void setRecordPeso(String recordPeso) {
        this.recordPeso = recordPeso;
    }

    public String getTipoMembresia() {
        return tipoMembresia;
    }

    public void setTipoMemebresia(String tipoMemebresia) {
        this.tipoMembresia = tipoMemebresia;
    }

    public String getEstado() {
        return estado;
    }

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

    public void entrenar(){

    }

}

```

Clase hija Coach:

```

package co.com.nttdata.gimnasio;

public class Coach extends Persona {

    private int idc;
    private String claseQueDicta;

    public Coach(int id, String nombres, String apellidos, String
direccion, String clave, int idc, String claseQueDicta) {
        super(id, nombres, apellidos, direccion, clave);
        this.idc = idc;
        this.claseQueDicta = claseQueDicta;
    }

    public int getIdc() {
        return idc;
    }

    public void setIdc(int idc) {
        this.idc = idc;
    }

}

```

```

    public String getClaseQueDicta() {
        return claseQueDicta;
    }

    public void setClaseQueDicta(String claseQueDicta) {
        this.claseQueDicta = claseQueDicta;
    }

    public void verAgendaClase(){

    }

    public void ajustarClases(){

    }

    public void darEntrenamiento(){

    }
}

```

Clase hija Administrador:

```

package co.com.nttdata.gimnasio;

public class Coach extends Persona {

    private int idc;
    private String claseQueDicta;

    public Coach(int id, String nombres, String apellidos, String
direccion, String clave, int idc, String claseQueDicta) {
        super(id, nombres, apellidos, direccion, clave);
        this.idc = idc;
        this.claseQueDicta = claseQueDicta;
    }

    public int getIdc() {
        return idc;
    }

    public void setIdc(int idc) {
        this.idc = idc;
    }
}

```

```

    public String getClaseQueDicta() {
        return claseQueDicta;
    }

    public void setClaseQueDicta(String claseQueDicta) {
        this.claseQueDicta = claseQueDicta;
    }

    public void verAgendaClase() {

    }

    public void ajustarClases() {

    }

    public void darEntrenamiento() {

    }
}

```

ContactoEmergencia

```

package co.com.nttdata.gimnasio;

public class ContactoEmergencia {

    private int id;
    private String nombres;
    private String apellidos;
    private String parentesco;
    private String telefono;

    public ContactoEmergencia() {

    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNombres() {
        return nombres;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }
}

```

```

    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public String getParentesco() {
        return parentesco;
    }

    public void setParentesco(String parentesco) {
        this.parentesco = parentesco;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
}

```

Clase

```

package co.com.nttdata.gimnasio;

public class ContactoEmergencia {

    private int id;
    private String nombres;
    private String apellidos;
    private String parentesco;
    private String telefono;

    public ContactoEmergencia() {

    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNombres() {

```



```
        return nombres;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public String getParentesco() {
        return parentesco;
    }

    public void setParentesco(String parentesco) {
        this.parentesco = parentesco;
    }

    public String getTelefono() {
        return telefono;
    }

    public void setTelefono(String telefono) {
        this.telefono = telefono;
    }
}
```