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**



**Product Backlog**

**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.

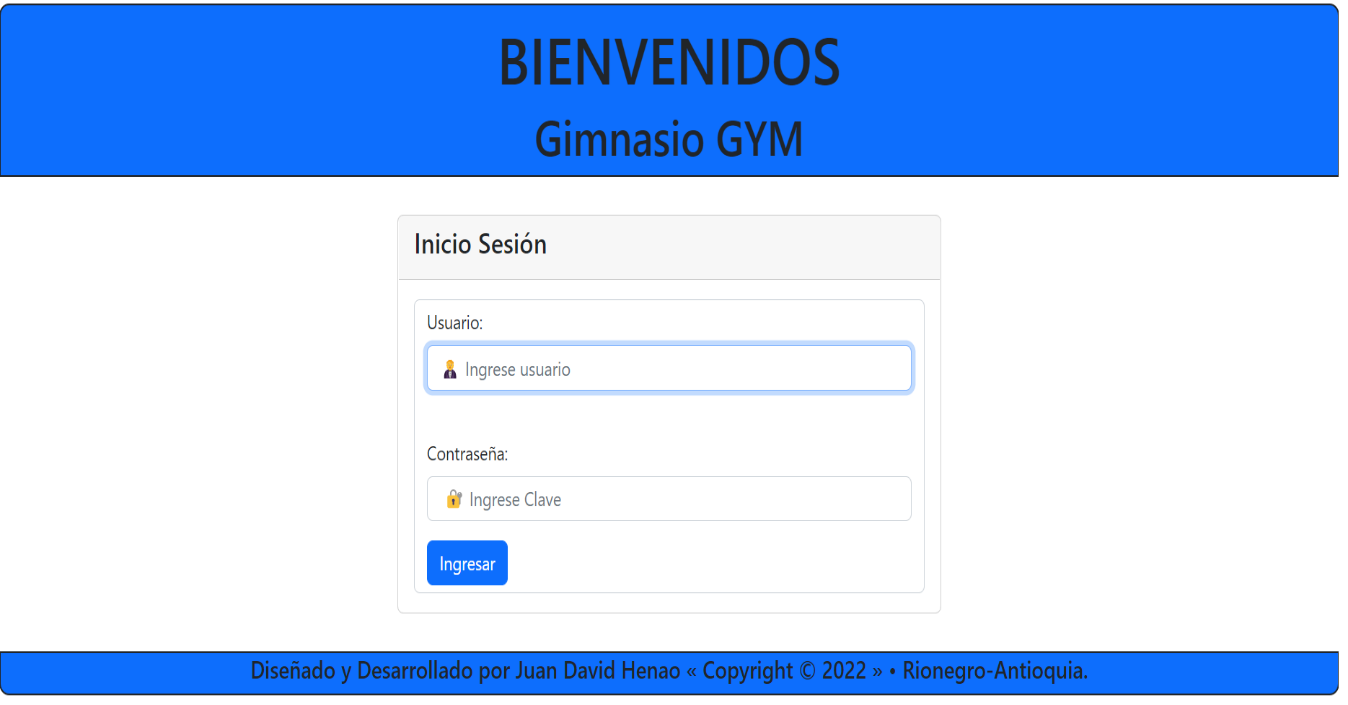
Ver tabla Sprint Backlog a continuación:

**Sprint Backlog**

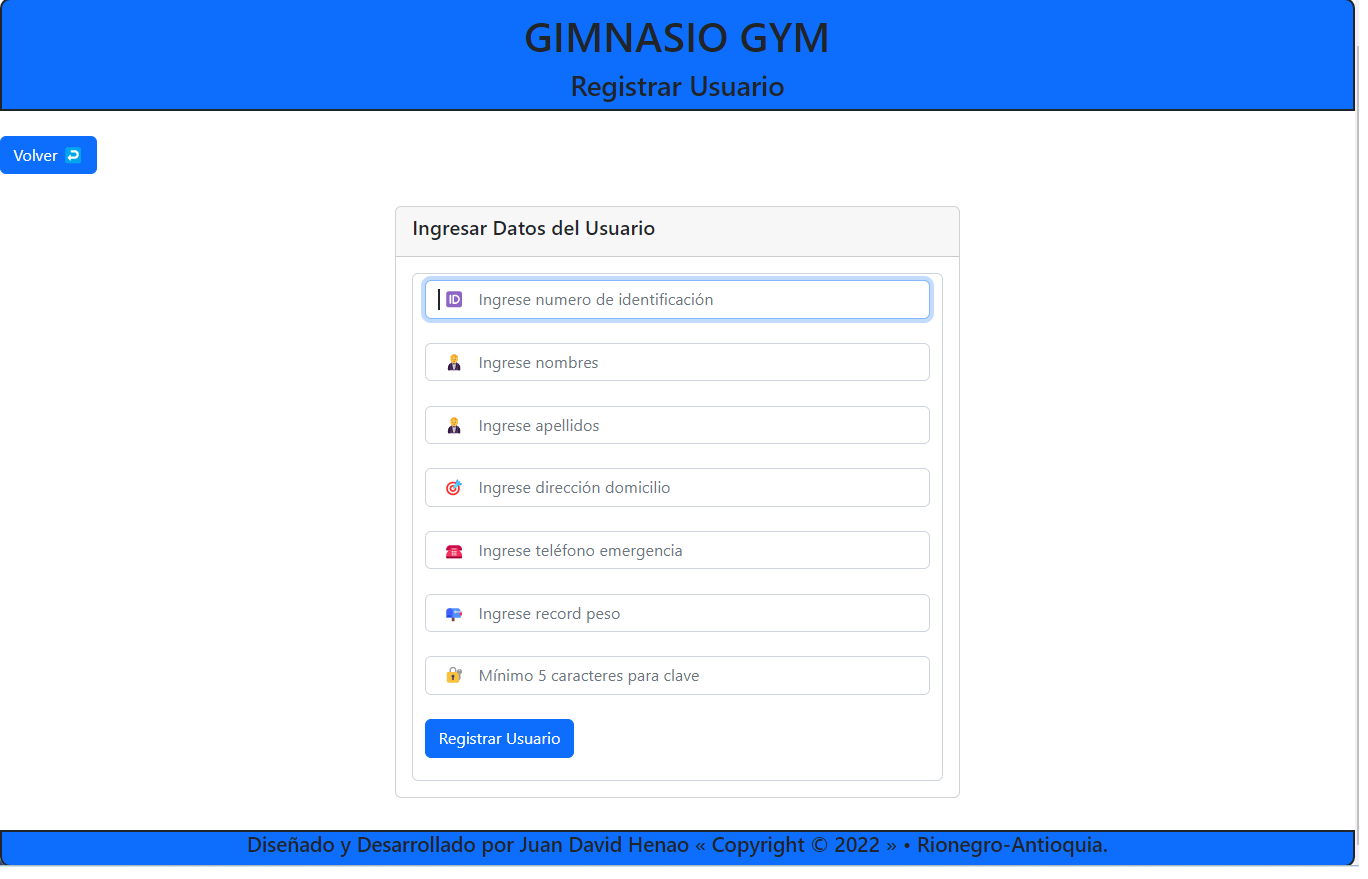
**Entregable**

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.

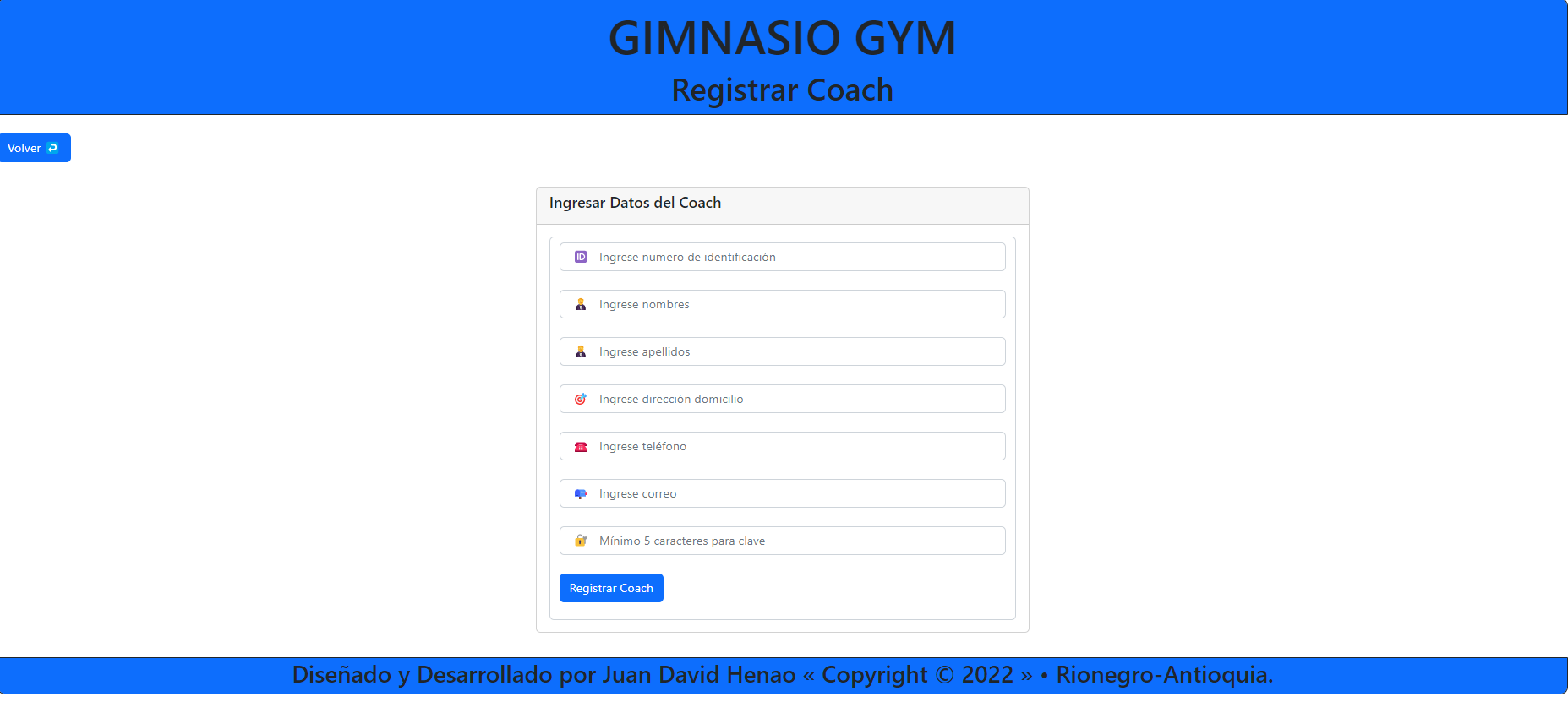
Este producto cuenta con las funciones necesarias para solucionar un problema específico.

Inicio de sesión:

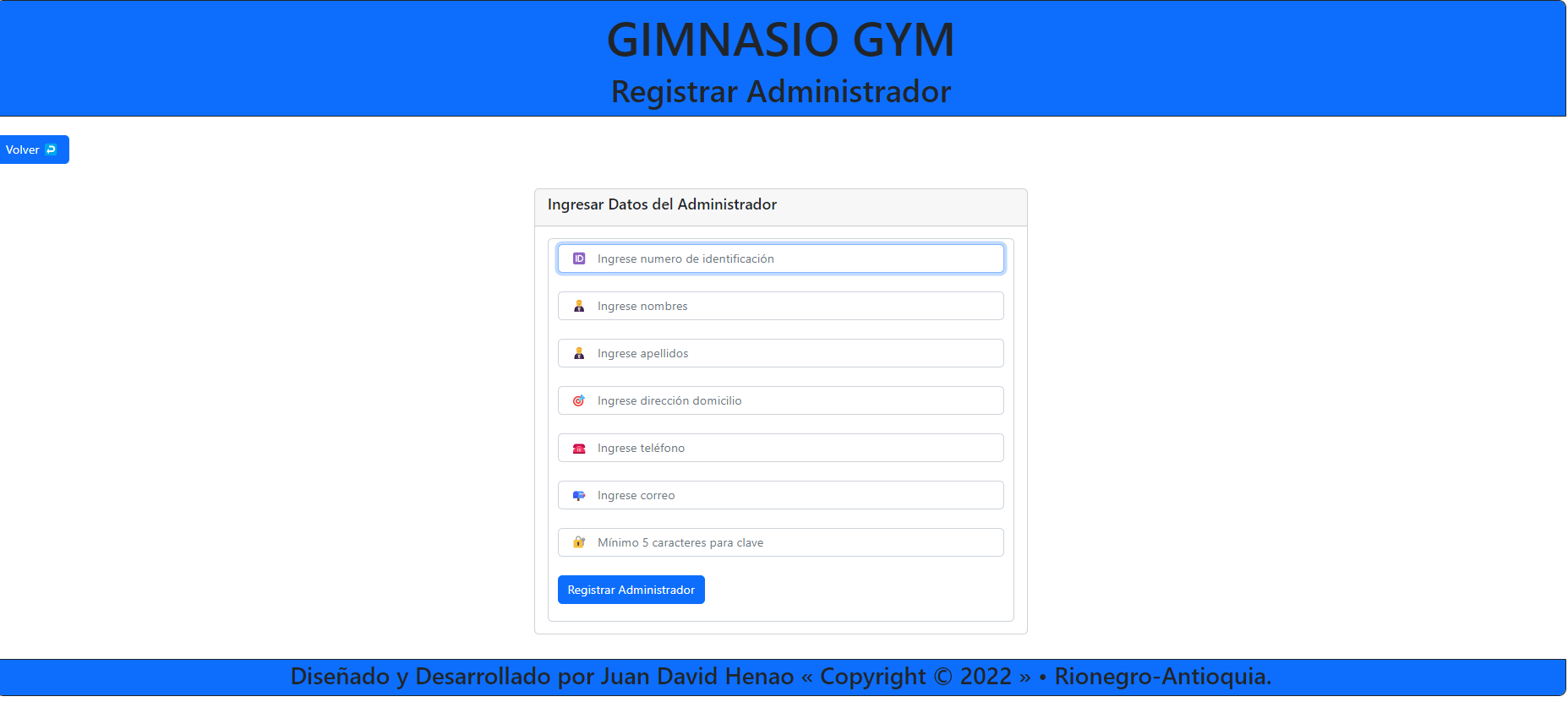
Registro usuario:

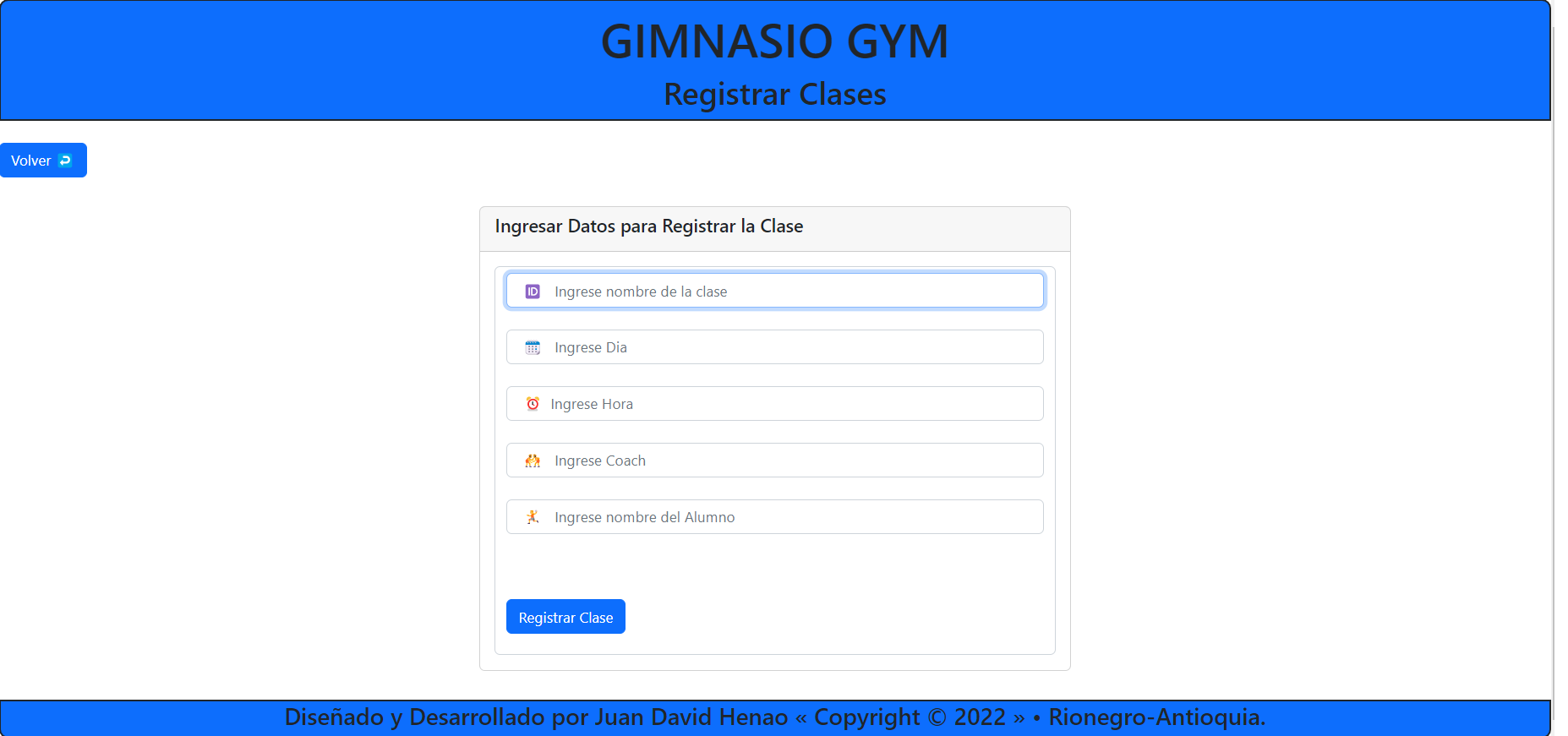


Registro coach:

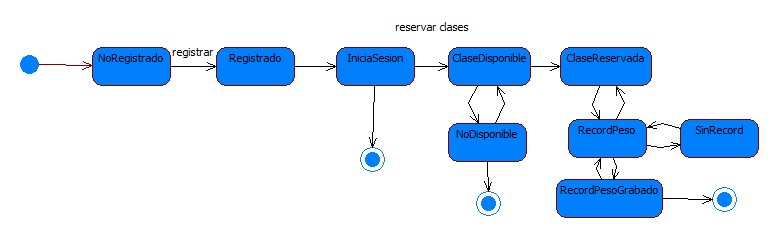


Registro Administrador:



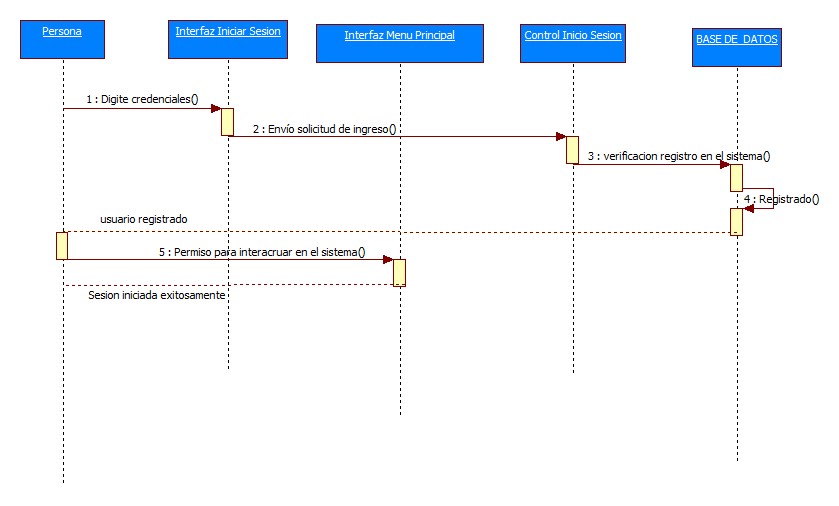
Registro clases:

Diagramas de estado



Diagramas de secuencia:

Secuencia Iniciar sesión



Secuencia Registro persona

Diagrama

Descripción generada automáticamente

Secuencia Registro Clases

Diagrama

Descripción generada automáticamente

Diagrama clases

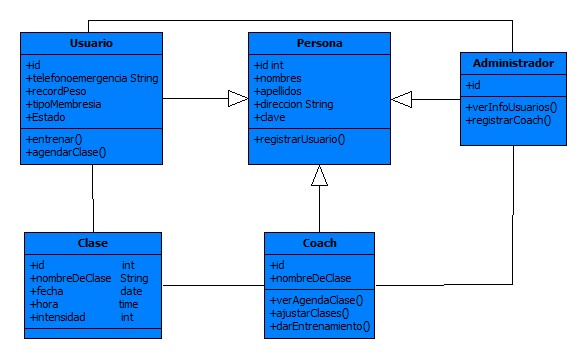
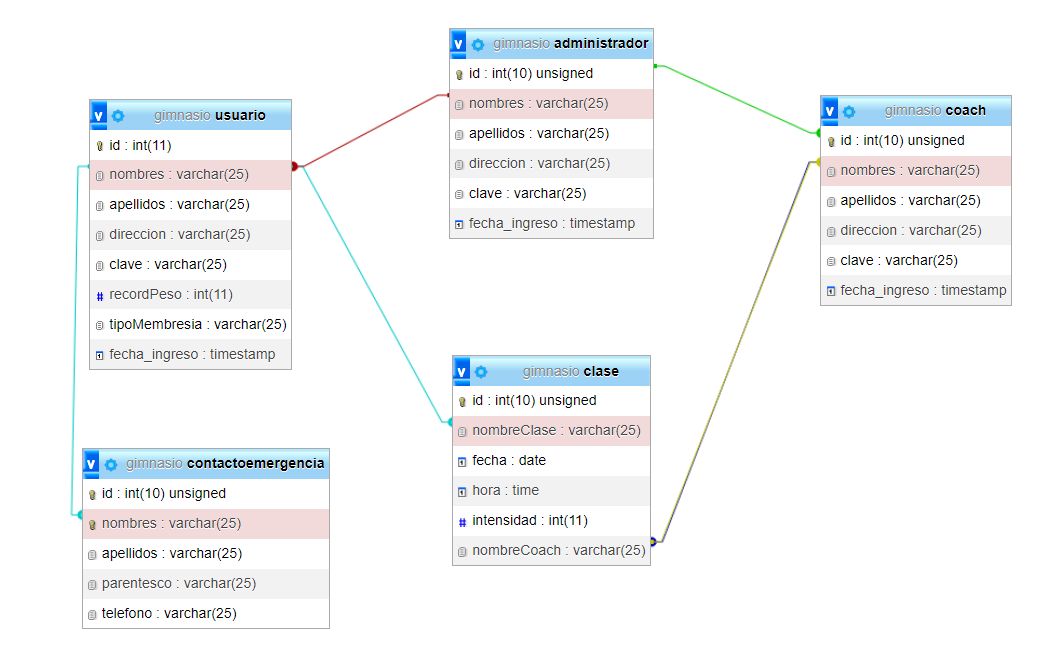


Diagrama entidad relación



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;  
 }  
}