

DISEÑO DE ACTIVIDADES MULTIPLATAFORMA

FITPLUS

Sistema Gestor de gimnasios

Sandra Rodríguez Espinosa

2018

ACADEMIA ASES

ÍNDICE

1. PORTADA	1
2. ÍNDICE	2
3. PLANTEAMIENTO	3
4. ANTECEDENTES	4
5. JUSTIFICACIÓN	5
6. CONTEXTUALIZACIÓN	6
7. DESARROLLO Y SERVICIOS	7
7.1. OBJETIVOS ESPECÍFICOS	
7.2. REQUERIMIENTOS TÉCNICOS	
7.3. REQUERIMIENTOS FUNCIONALES	
7.4. TECNOLOGÍAS IMPLEMENTADAS	
7.5. ESQUEMAS	
7.5.1. PERT	
7.5.2. GANT	
7.5.3. E-R	
7.5.4. ESQUEMA LÓGICO	
8. SERVICIOS Y MODULOS	16
8.1. APLICACIÓN JAVA	
8.2. APLICACIÓN PHP	
9. RELACIÓN ENTRE LOS MÓDULOS	25
9.1. APLICACIÓN JAVA	
9.2. APLICACIÓN PHP	
10. TESTER	31
10.1. APLICACIÓN JAVA	
10.2. APLICACIÓN PHP	
11. BIBLIOGRAFÍA	36

PLANTEAMIENTO

En el siguiente proyecto que se vamos a plantear , vamos a desarrollar una aplicación de sistema de gestión de gimnasios . El cual contiene las siguientes partes :

- 1- Por una parte tendremos la parte administrativa , donde el administrador del gimnasio , podrá dar de alta , eliminar nuevos usuarios y además modificarlos . Además podrá filtrar , la gente que ha pagado su recibo mensual o si necesita buscar algún cliente en concreto para poder comprobar algún dato concreto.
- 2- Por otra dispondremos de una parte PHP , con una plantilla HTML , donde el cliente podrá acceder a su cuenta de gimnasio y podrá ver todas los ejercicios y series que ha realizado y le queda por hacer.
- 3- Ambas se conectarán a la misma base de datos , llamada “proyecto Final” , donde estarán todos los datos de clientes , ejercicios y logins correspondientes.

ANTECEDENTES

Una de los antecedentes más importantes que me dio la idea a iniciar un proyecto de este tipo , fueron las necesidades que veía en los gimnasios donde yo iba : VIVA GYM , FORMA entre otros .

Como clienta ,el diseño adquirido me parecía poco intuitivo , además de la dificultad para el cliente de darse de alta , cada vez que iba el cliente a darse de alta , tenía que rellenar de nuevo todos los datos , eso retrasaba mucho más el trabajo a nivel del administrador y a nivel del cliente , era pesado.

No obstante , la búsqueda de información también ha sido adquirida de los siguientes lugares :

- Otras aplicaciones web sobre gimnasios para diseño de la parte java.
- Guía de otros ejercicios en clase.
- Github.
- Cuestiones varias por Google.

JUSTIFICACIÓN

En primer lugar ,mi elección del proyecto ha sido un *sistema gestor* porque creo que es algo útil , no solo para gimnasios , sino para cualquier tipo de centro que se tenga que dar de alta un cliente , este sistema es necesario para cualquier centro donde haya un administrador , bien sea gimnasios , centros de estética , peluquerías , etc....

Hoy en día , el tema de gimnasios es muy popular , el culto al cuerpo se ha ido haciendo cada vez más relevante para nuestra sociedad , nos hemos convertido en una generación donde el deporte , es algo imprescindible para nuestro bienestar personal y físico , por ello el proyecto se ha centrado más en gimnasios.

Además, es algo sencillo y más rápido de poder desarrollar , dada el límite de horas para el proyecto.

CONTEXTUALIZACIÓN

Actualmente las aplicaciones web han ido causando una relevante importancia en la sociedad , ahora prácticamente todo , se puede gestionar a través de webs , es más , ahora TODO se puede gestionar a través de aplicaciones móvil.

Esto ha influido positivamente en la sociedad , ya que hoy en día todo esta muy informatizado y en general se genera más rapidez a la hora de cualquier actividad que quieras realizar , por ejemplo ver tu cuenta de banco , acceder a tus datos , envíos de documentos desde una plataforma , en general , es una avance para la sociedad. Pero no hace falta hacerlo desde la web sino desde el móvil , todo es accesible , no hay necesidad de tener un ordenador.

Es por eso que he utilizado una plantilla *responsive*, indistintamente de dónde entre el cliente ,web o móvil, no tendrá problemas de visualización de sus ejercicios en su *área de cliente*.

El área donde se va a mover este tipo de aplicación java ,es más en gimnasios , cualquier tipo de gimnasio donde un cliente se tenga que dar de alta.

DESARROLLO SERVICIOS

- **OBJETIVOS ESPECÍFICOS**

- Reforzar los conocimientos Java y conocimientos PHP.
- Aprender a utilizar una plantilla *responsive*.
- Hacer un diseño más intuitivo y sencillo.

- **REQUERIMIENTOS TÉCNICOS**

- Xampp (Servidor Apache y MySQL) para conectar base de datos Y Netbeans
- Netbeans IDE 8.2– parte java Swing y PHP
- Sqlyog Community Edition.
- Plantilla html .

- **REQUERIMIENTOS FUNCIONALES**

- El administrador podrá registrar al nuevo cliente en base de datos, con todos los datos personales del cliente.
- El administrador podrá eliminar al nuevo cliente en base de datos.
- El administrador podrá visualizar todos los clientes del gimnasio.
- El administrador deberá loguearse con su usuario y contraseña para poder demostrar que es la persona responsable para poder acceder a la modificación de la base de datos.
- La contraseña de administrador tendrá que validarse en la base de datos.
- El administrador tendrá valor NULL en el nivel , si tiene nivel no dejará entrar en la aplicación.
- Si el usuario y contraseña del administrador no son válidos , la aplicación se bloqueará y solo podrá salir el administrador.
- El administrador elegirá un nivel para el usuario en el momento del alta del cliente.
- Mantenedor de usuarios.
- La aplicación web deberá dejar de poder entrar en su sesión a través de su usuario y password y visualizar sus rutinas diarias dependiendo del nivel.
- Es necesario que el usuario pertenezca a un nivel dentro de la aplicación para poder acceder a estas .

- **TECNOLOGÍAS IMPLEMENTADAS**

- Java versión : 1.8.

Java es uno de los lenguajes a día de hoy con un alto reconocimiento en el mercado del mundo de la tecnología . Es por ello que he decidido sumergirme en este proyecto con este lenguaje. Cada vez Java , sigue creciendo más y más.

La demanda de expertos en Java es de las más altas , a mi me sirve para reforzar los conocimientos y tener más opción a nuevos aires en el empleo laboral e inspiraciones.

- PHP versión: 1.8.

Después de investigar más sobre esta tecnología ,actualmente PHP está presente, aproximadamente, en el 24% de las páginas existentes, una cifra relevante. Es un lenguaje abierto y libre, con una línea de aprendizaje baja , sus entornos de aprendizaje son de rápida y fácil configuración , tiene fácil acceso a la base de datos , además de una comunidad muy grande .Reforzar este lenguaje , ha sido otro de mis objetivos , nunca se sabe si quedará obsoleto o seguirá adelante , pero si siguen poniendo más actualizaciones sobre este lenguaje , está claro que es un lenguaje que no morirá en el camino.

- MySQL GUI 8.05.

Una de las cosas importantes sobre el proyecto ha sido la elección de las bases de datos , aunque hay distintas bases de datos , he elegido MySQL , primero porque es la base de datos que tengo más conocimiento , ya que es la que hemos trabajado durante estos dos años de formación .

Uno de los beneficios de la elección es que es multiplataforma , así no habrá problemas de acceso. Por supuesto es muy fácil encontrar cualquier tipo de ayuda sobre internet , cualquier duda puedes encontrar de forma rápida una solución , *querys*(consultas) , errores, etc ... Además tiene una curva de aprendizaje fácil para el usuario que la implementa.

- **ESQUEMAS**

- PERT

- A . Elección del Proyecto (IDEAS).
- B. Análisis de la Gestión Proyecto.
- C. Base de Datos.
- D. Parte Java (Diseño - Frontend).
- E. Parte Java (Funcionalidad - Frontend).
- F . Parte PHP (Diseño - Backend) .
- G. Parte PHP(Funcionalidad - Backend).

ACTIVIDAD	PRECEDE
A	-
B	A
C	B
D	C
E	D
F	C
G	E

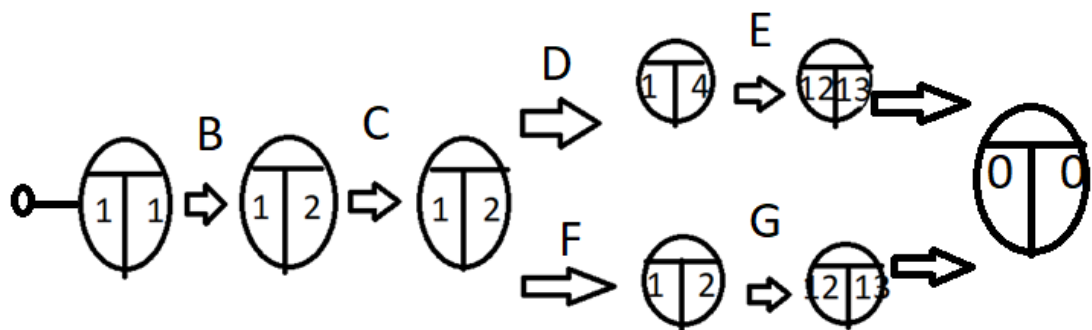


Figura 1 : DIAGRAMA DE PERT

FIGURA 1 :

El diagrama de Pert nos permite dirigir la programación de un proyecto , qué tareas , cuando y el objetivo que queremos lograr.

He dividido el proyecto en 7 apartados :

El primero(Actividad A) es la elección del proyecto , contiene análisis , búsqueda de información , valoración relación-tiempo,... que no precede de ninguna actividad anterior por lo que en el esquema es donde se inicia la actividad con un circulo en blanco .

Cuando ya hemos decidido el proyecto , el segundo(Actividad B) es el análisis de Gestión de Proyecto , es una actividad lineal que precede de la actividad A.

En tercer lugar Base de datos(Actividad C) , creación , análisis etc... precediendo de la actividad B.

En cuarto lugar Parte JAVA(Actividad D) , el diseño ,que precede de la actividad C.

En quinto lugar Parte JAVA(Actividad E) ,la funcionalidad, precediendo de la actividad D.

En cuarto lugar Parte PHP(Actividad F) , el diseño ,que precede de la actividad C

En cuarto lugar Parte PHP(Actividad G) , su funcionalidad ,que precede de la actividad F.

A continuación se procede a graficar el Diagrama PERT. Normalmente se comienza de izquierda a derecha. Se emplean nodos para denotar inicio y fin de las actividades, líneas rectas horizontales (preferiblemente) para representar las actividades e identificadas con la etiqueta en la parte superior y la duración en la parte inferior.

Una vez que está graficado el PERT, se calculan los tiempos de ejecución de las actividades. Se comienza desde la izquierda, colocando el valor 0 en el cuadro izquierdo del primer nodo, y se va sumando el tiempo de duración de cada actividad para ir acumulando el tiempo.

Al terminar de calcular los tiempos de ejecución, se procede a calcular los tiempos de holgura, de derecha a izquierda. Se coloca en el cuadro derecho del último nodo el tiempo total del proyecto, y se va restando el tiempo de duración de cada actividad, para ir disminuyendo el tiempo. Cuando existan dos o más acumulados se toma el acumulado de menor duración.

.

- GANT

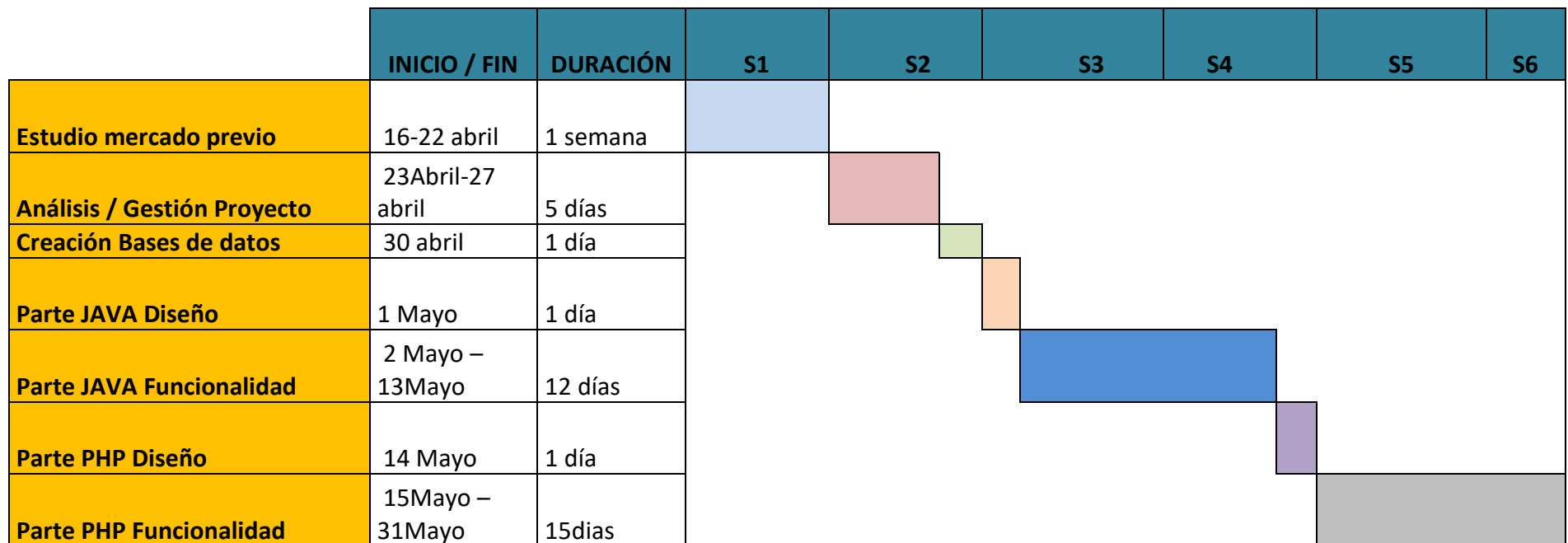


Figura 2: DIAGRAMA DE GANT.

FIGURA 2 :

En el *diagrama de Gant* , podemos observar las diferentes tareas del proyecto dividido en diferentes términos de tiempo , cada fila representada por un espacio de tiempo , y señalado en cada semana por colores .

La primera semana me dediqué al estudio de mercado , necesidades , demanda de cliente , y posibles ideas de proyecto.(1 Semana)

Cuando ya tuve decidido la idea , viene la organización de todo , en que tecnología la iba a implementar , que sección iba a ser en JAVA que sección iba a ser en PHP y como iba a definir mi Base de Datos.(5 días)

A raíz de ahí , me puse a implementar lo que sería el diseño(1 día) y a continuación darle funcionalidad(12 días). Por supuesto siempre algo orientativo ya que siempre se puede hacer mejoras sobre la misma aplicación depende de que necesidades tuviera a medida que iba avanzando en el proyecto.

Más tarde , ya implementé la parte PHP de diseño(1 día) , cogido de una plantilla HTML , luego lo más relevante de esta parte ,su funcionalidad .(15 días)

La parte del *testing* (prueba) unitario , modular y global , me llevo dos días más (último fin de semana) para poder probarla.

- **ESQUEMA E-R**

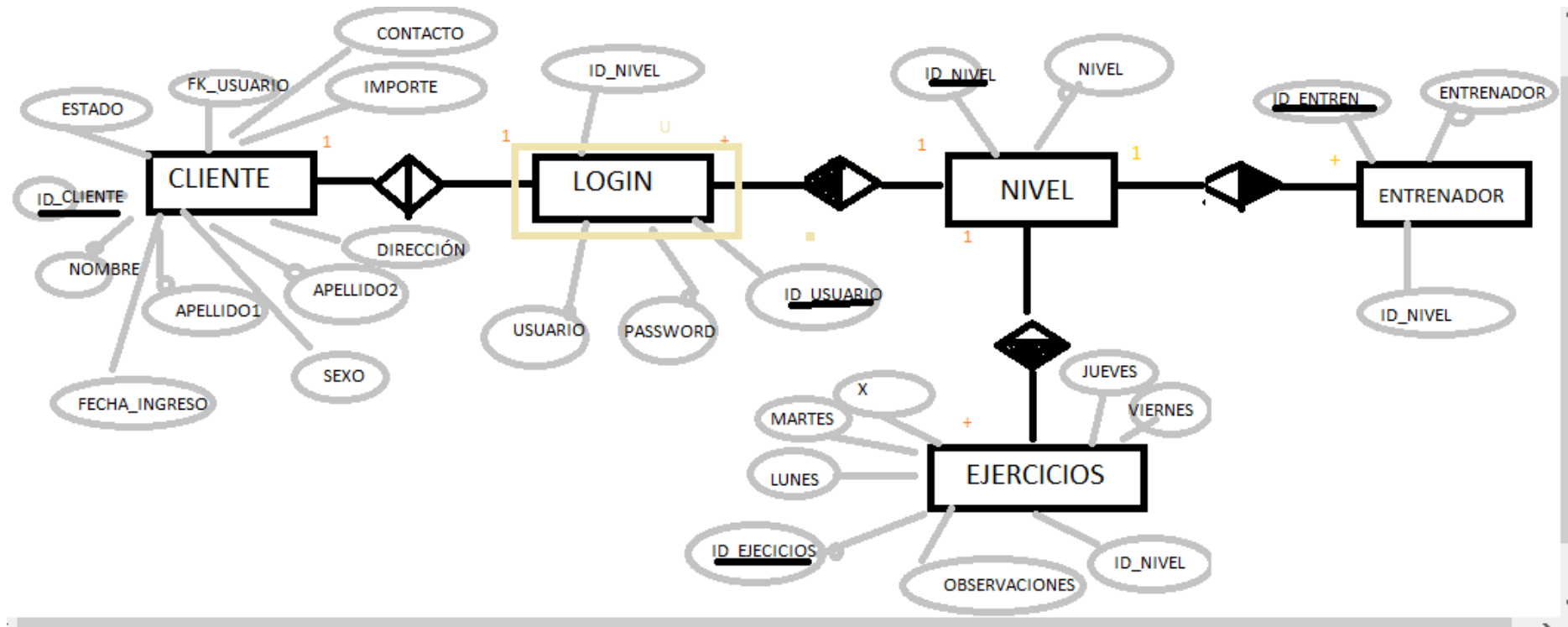


Figura 3 : ESQUEMA ENTIDAD – RELACIÓN

FIGURA 3:

En el siguiente esquema conceptual , se muestra todas las relaciones de la base de datos, y sus tablas. Los cuadros son las tablas , con todos sus campos asociados a esa tabla.

Un CLIENTE tiene un LOGIN (Usuario), y un LOGIN solo puede pertenecer a un USUARIO , por lo tanto tabla Login tiene un valor UNIQUE(unicidad).

Un LOGIN pertenece a un NIVEL , y un NIVEL puede pertenecer a muchos clientes , por lo que la Foreign Key , aparece en la tabla Login , por eso aparece coloreado en negro.

Un NIVEL puede tener muchos EJERCICIOS , y un EJERCICIOS solo puede tener un nivel, por lo que la Foreign Key , estaría en la tabla Ejercicios.

Así la relación quedaría de la siguiente manera , :

-CLIENTE- LOGIN (relación 1-1) UNIQUE .

-LOGIN – NIVEL (relación MUCHOS(+) - 1).

- NIVEL- EJERCICIOS(relación 1-MUCHOS(+)).

- NIVEL – ENTRENADOR (relación 1 - +)

A continuación , pasaremos al Esquema Lógico , dónde hablaremos de todos los campos de cada tabla y de sus características con sus FK.

- **ESQUEMA LÓGICO**

- **CLIENTE** (int ID_CLIENTE ,varchar(11) NOMBRE, varchar(11)APELLIDO1,varchar(11)APELLIDO2, varchar(11)SEXO,typeint(1,0)ESTADO, varchar(20)DIRECCIÓN , varchar(11)FECHA_INGRESO ,ID_USUARIO)

ID_CLIENTE : Primary Key (Clave Primaria)PK;

NOMBRE : Valor not Null (Valor no nulo)VNN;

APELLIDO1: Valor not Null (Valor no nulo)VNN;

APELLIDO2: Valor not Null (Valor no nulo)VNN;

SEXO: Valor not Null (Valor no nulo)VNN;

USUARIO: (fk.usuario -> LOGIN(usuario));

- **LOGIN**(ind ID_USUARIO , varchar(11)USUARIO, varchar(11)PASSWORD)UNIQUE

ID_USUARIO: Primary Key (Clave Primaria)PK;

USUARIO: Valor not Null (Valor no nulo)VNN;

PASSWORD: Valor not Null (Valor no nulo)VNN;

- **NIVEL** (int ID_NIVEL, varchar(11)NIVEL)

ID_NIVEL: Primary Key (Clave Primaria)PK;

- **EJERCICIOS**(int ID_EJERCICIO, varchar(11)EJERCICIO, varchar(20)OBSERVACIONES,int ID_NIVEL)

ID_EJERCICIO : Primary Key (Clave Primaria)PK;

EJERCICIO: Valor not Null VNN;

ID_NIVEL : (id_nivel - > NIVEL(id_nivel));

- **ENTRENADOR** (int ID_ENTRENADOR, varchar(11)ENTRENADOR)

ID_ENTRENADOR: Primary Key (Clave Primaria)PK;

ENTRENADOR : Valor not Null VNN;

ID_NIVEL : (id_nivel - > NIVEL(id_nivel));

- **SERVICIOS / MÓDULOS**

- APLICACIÓN JAVA.

MÓDULO LOGIN
1. Permite ingresar usuario y password al administrador para poder acceder a la plataforma .
2. Al pulsar Cancelar sale de la aplicación y no deja entrar en la plataforma .
3. Si se ingresa el usuario o password mal , la aplicación se bloqueará y no permitirá poder seguir con el proceso.

MÓDULO MODELO
1. Permitirá ingresar los datos personales del cliente nuevo: nombre , apellidos... y dar alta en la base de datos todos los datos personales del nuevo cliente.
2. Permitirá acceder a la muestra de todos los clientes .

MÓDULO CLIENTES
1. Se visualiza en la tabla todos los datos de clientes del gimnasio. Nombre , Apellido1 , Apellido2 , Estado si esta pagado o no , y su dni.
2. Tendrá filtros para buscar los clientes determinados , por nombre , apellido , apellido2 , dni y si se ha pagado o no la mensualidad.
3. Seleccionando al cliente , modificando el dato que se quiera modificar y pulsando MODIFICAR CLIENTE , se modificará directamente en la base de datos.
4. Seleccionando al cliente y pulsando ELIMINAR permite también eliminar a un cliente determinado de la base de datos de la aplicación.

MÓDULO MySQL
1. Contendrá un modulo con el que se conectará a la base de datos y otro que llamaremos siempre para la desconexión , por cada vez que se abra , cerraremos la conexión.
2. Contendrá un método al que llamaremos cada vez que

queramos insertar a la base de datos un nuevo cliente , con la query (consulta)correspondiente.
3. Contendrá un método al que llamaremos cada vez que queramos eliminar de la base de datos un cliente , con la query (consulta)correspondiente.
4. Contendrá un método al que llamaremos cada vez que queramos modificar de la base de datos un cliente , con la query (consulta)correspondiente.
5. Contendrá un método para saber si el administrador , existe en la base de datos ,tal administrador para poder comprobar más adelante si es correcto el administrador.

- **RELACION ENTRE MÓDULOS E INTEGRACIÓN SOBRE ELLOS.**

- *Conexión a la base de datos*

```
public void conexion (String user , String password ) throws SQLException{

    try {

        Class.forName("com.mysql.jdbc.Driver");

        Conexion=DriverManager.getConnection("jdbc:mysql://localhost:3306/proyecto-final", user, password);

        // JOptionPane.showMessageDialog(null, "Conectado a la base de datos");

    } catch (ClassNotFoundException ex) {

        Logger.getLogger(MySQL.class.getName()).log(Level.SEVERE, null,ex);

    }

}
```

- *Desconexión a la base de datos*

```
public void desconexion (){  
  
    try {  
  
        Conexion.close();  
  
        //JOptionPane.showMessageDialog(null, "Desconectado de la base de  
        datos");  
  
    } catch (SQLException ex) {  
  
        Logger.getLogger(MySQL.class.getName()).log(Level.SEVERE, null,  
ex);  
  
    }  
  
}
```

MÓDULO LOGIN – MÓDULO MODELO : Al pulsar en ACEPTAR , siempre y cuando el usuario sea correcto para entrar en la aplicación desaparecerá la ventana del LOGIN y entrará en el Módulo MODELO.

Figura 4 : PANTALLAZO DE LOGIN.

```

public void compruebaAdmin() {
    try {
        db.conexion("root", "");
        String usuario = this.jTextField1.getText();
        String password = this.jTextField2.getText();
        if(db.compruebaBDadmin(usuario, password)) {
            this.dispose();
        } else {
            JOptionPane.showMessageDialog(null, "El usuario/password no
                existen.");
        }
        db.desconexion();
    } catch (Exception ex) {

        Logger.getLogger(Identificacion.class.getName()).log(Level.SEVERE, null, ex);
        db.desconexion();
    }
}

```

The screenshot displays the FITPLUS application interface. At the top, there is a navigation bar with 'Inicio' and 'Salir' links. The main title 'FITPLUS' is centered. Below the title, the form is organized into two columns. The left column contains fields for 'NOMBRE' (Sandra), 'APELLIDO 1' (Rodriguez), 'APELLIDO 2' (Espinosa), 'DNI' (0987665N), 'FECHA NACIMIENTO' (1988-02-01), 'TELÉFONO' (65234567), 'DIRECCIÓN' (pintor sorolla n29), and 'SEXO' (radio buttons for Femenino and Masculino). The right column contains fields for 'Usuario' (sandy), 'PASSWORD' (12345), 'FECHA INGRESO' (2000-08-01), 'IMPORTE' (230), 'ESTADO' (dropdown menu showing PAGADO), and 'NIVEL' (dropdown menu showing INICIAL). At the bottom right, there are two buttons: 'DAR DE ALTA' and 'MOSTRAR CLIENTES'.

Figura 5 : PANTALLAZO DE MODELO.

CLIENTES

NOMBRE: USUARIO:

APELLIDO 1: APELLIDO 2:

DNI:

ESTADO: NIVEL:

USUARIO	NOMBRE	APELLIDO 1	APELLIDO 2	DNI	ESTADO	NIVEL
sandyesyy	Sandra	Rodriguez	Espinosa	0987665N	PAGADO	INTERMEDIO
sandyqqq	Sandra	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL
sandyqqqqee	Sandra	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL
sandyeeee	Sandra	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL
aaaaa	Sandra	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL
sandyeeeeeee	Sandra	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL
fede	fede	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL
sandyeee	fede	Rodriguez	Espinosa	0987665N	PAGADO	INICIAL

Figura 6 : PANTALLAZO DE CLIENTES.

FIGURA 5 – 6 :

MÓDULO MODELO – MÓDULO CLIENTES : El módulo modelo al pulsar sobre **MOSTRAR CLIENTES**, aparecerá el módulo de clientes dados de alta.

```
private void BotonAlta(java.awt.event.ActionEvent evt) {
    try {
        db.conexion("root", "");
        String nombre = this.jTnombre.getText();
        String apellido = this.jTapellido1.getText();
        String apellido2 = this.jTapellido2.getText();
        String dni = this.jTdni.getText();
        String fecha = this.jTfnacimiento.getText();
        String direccion = this.jTdireccion.getText();
        String telefono = this.jTtelefono.getText();
        String fechaIngreso = this.jTfecIngreso.getText();
        String importe = this.jTimporte.getText();
        String estado = (String)this.jComboBox1.getSelectedItem();
        String usuario = this.jTusuario.getText();
        String password = this.jTpassword.getText();
        String nivel = (String)this.jComboBox2.getSelectedItem();

        String sexo= "";
        // seleccion del radio
        if(this.jRFem.isSelected()){
```

```

        sexo = "femenino";
    }else if(this.jRMas.isSelected()){
        sexo = "masculino";
    }

    int id_nivel = 1;
    if(nivel.toUpperCase().equals("INTERMEDIO")) {
        id_nivel = 2;
    } else if(nivel.toUpperCase().equals("AVANZADO")) {
        id_nivel = 3;
    }

    if(nombre.trim().equals("") || apellido.trim().equals("") ||
    apellido2.trim().equals("") ||
        dni.trim().equals("") || fecha.trim().equals("") ||
    direccion.trim().equals("") ||
        direccion.trim().equals("") || telefono.trim().equals("") ||
    fechaIngreso.trim().equals("") ||
        importe.trim().equals("") || usuario.trim().equals("") ||
    password.trim().equals("") ||
        sexo.equals("")) {
        JOptionPane.showMessageDialog(null, "Debe rellenar todos los
campos.");
    } else {
        db.insertar(nombre, apellido, apellido2, dni, sexo, fecha, direccion
,telefono, fechaIngreso,importe,estado,usuario,password,id_nivel);
    }
    db.desconexion();

    } catch (SQLException ex) {
        Logger.getLogger(Modelo.class.getName()).log(Level.SEVERE, null,
ex);
        JOptionPane.showMessageDialog(null, "Cliente con usuario " +
this.jTusuario.getText() + " ya existe");
    }
}

private void BotonMostrar(java.awt.event.ActionEvent evt) {
    try {
        cli = new Clientes(null,true);
    } catch (SQLException ex) {
        Logger.getLogger(Modelo.class.getName()).log(Level.SEVERE, null,
ex); }
    cli.setVisible(true);
}

```

- *Select FILTROS de los clientes*

En el siguiente código , recogemos en una query todos los datos de todas las tablas con varios **INNER JOIN**, y a partir de ahí, si queremos filtrar por nombre, apellidos, etc, preguntara con un **if** si hay dentro texto y si es así , se añadirá al **WHERE** de la query.

Para finalizar la query tenemos una variable finquery para el caso en que haya algún campo con texto y necesitamos que la query no termine en AND, por lo cual se filtrará además todos los c.id_cliente > 0 para asi poder ejecutar bien la query con su WHERE y campos filtrados.

// busqueda en bd de filtros

```
public ArrayList filtros(String nombre ,String apellido ,String
apellido2,String dni,String estado, String nivel, String usuario){
    valores = new ArrayList();
    try {
        String query = "SELECT c.usuario as usuario, c.nombre as nombre,
c.apellido as apellido, "
            + "c.subapellido as subapellido, c.dni as dni, c.estado as estado,
n.nivel as nivel FROM clientes c "
            + "INNER JOIN login l ON (c.usuario = l.usuario) "
            + "INNER JOIN nivel n ON (l.id_nivel = n.id_nivel) ";
        String finquery = "";
        if(!nombre.trim().equals("") || !apellido.trim().equals("") ||
!apellido2.trim().equals("")) ||
            !dni.trim().equals("") || !estado.equals("TODOS") ||
!nivel.equals("TODOS") || !usuario.trim().equals("")) {
            query += " WHERE ";
            finquery = " c.id_cliente > 0";
        }
        if(!usuario.trim().equals("")) {
            query += " c.usuario LIKE '%" + usuario + "%' AND ";
        }
        if(!nombre.trim().equals("")) {
            query += " c.nombre LIKE '%" + nombre + "%' AND ";
        }
        if(!apellido.trim().equals("")) {
            query += " c.apellido LIKE '%" + apellido + "%' AND ";
        }
        if(!apellido2.trim().equals("")) {
            query += " c.subapellido LIKE '%" + apellido2 + "%' AND ";
        }
        if(!dni.trim().equals("")) {
```

```

        query += " c.dni LIKE '%" + dni + "%' AND ";
    }
    if(!estado.equals("TODOS")) {
        query += " c.estado = '" + estado + "' AND ";
    }
    if(!nivel.equals("TODOS")) {
        int id_nivel = 1;
        if(nivel.toUpperCase().equals("INTERMEDIO")) {
            id_nivel = 2;
        } else if(nivel.toUpperCase().equals("AVANZADO")) {
            id_nivel = 3;
        }
        query += " l.id_nivel = " + id_nivel + " AND ";
    }
    if(!finquery.equals("")) {
        query += finquery;
    }
}

```

- **APLICACIÓN PHP**

- **SERVICIOS / MÓDULOS**

MÓDULO INDEX
1.En la pagina permite visualizar las distintas actividades que se pueden realizar en el gimnasio.
2. También se verá los entrenadores que las dirigen y las diferentes actividades que se realizan en el gimnasio.
3. Permite acceder al area de CLIENTES para poder ver su rutina diaria dependiendo del nivel de usuario , que ingresado el administrador.

MÓDULO LOGIN
1.Permite ingresar usuario su usuario y password para poder ver su rutina diaria , depende del nivel de usuario.
2. Si el usuario ingresa mal la password o usuario , directamente la aplicación te avisará de que no existe en la base de datos y no podrá acceder a la aplicación ni a ningún tipo de acción dentro de la plataforma.

MÓDULO RUTINA
1.Permite visualizar todos los ejercicios del usuario correspondiente con su nivel , por día de la semana , actividades realizadas y en las observaciones estará las sesiones y repeticiones de cada ejercicio.

- **RELACION ENTRE MÓDULOS E INTEGRACIÓN SOBRE ELLOS.**

- *Conexión a la base de datos*

```
<?php

$conexion = mysqli_connect("localhost", "root",
"", "proyectofinal");

if (!$conexion) {

    //echo "Error al conectar";

} else {

    //echo "conectado a BD";

}

?>
```

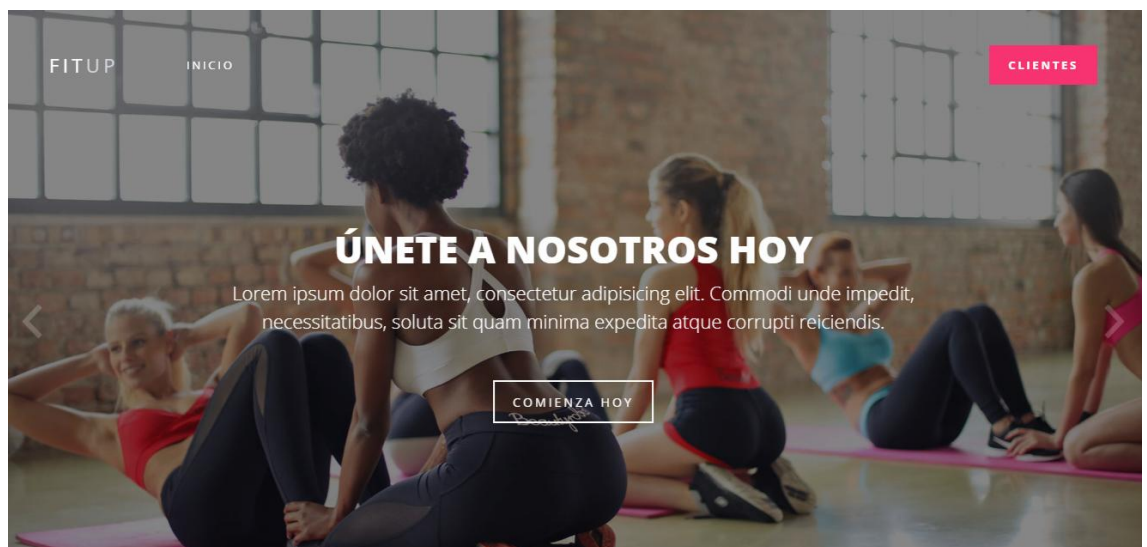


Figura 7 : PANTALLAZO DE INDE

- Creación tabla de entrenadores

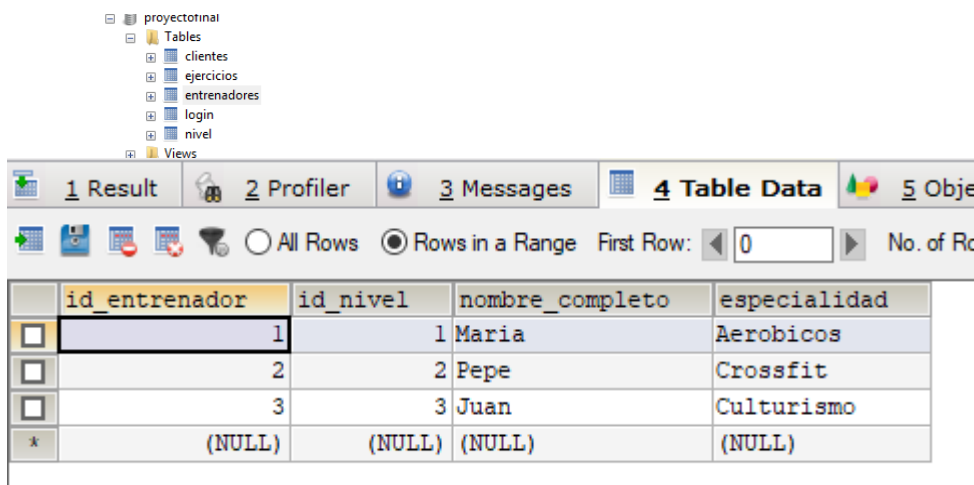
```
CREATE TABLE `entrenadores` (  
  `id_entrenador` int(11) NOT NULL,  
  `id_nivel` int(11) NOT NULL,  
  `nombre_completo` varchar(400) NOT NULL,  
  `especialidad` varchar(50) NOT NULL  
)
```

- Inserción a la tabla entrenadores

```
INSERT INTO `entrenadores` (`id_entrenador`, `id_nivel`, `nombre_completo`,  
`especialidad`) VALUES(1, 1, 'Maria', 'Aerobicos'),(2, 2, 'Pepe', 'Crossfit'),(3, 3, 'Juan',  
'Culturismo');
```

- Relación con tabla nivel

```
ALTER TABLE `entrenadores`  
  
ADD PRIMARY KEY (`id_entrenador`),  
  
ADD KEY `fk_nivel` (`id_nivel`);
```



The screenshot shows a database management interface with a tree view on the left containing 'proyectofinal', 'Tables', 'clientes', 'ejercicios', 'entrenadores', 'login', 'nivel', and 'Views'. The main window displays the 'entrenadores' table data. The table has four columns: 'id_entrenador', 'id_nivel', 'nombre_completo', and 'especialidad'. The data rows are:

	id_entrenador	id_nivel	nombre_completo	especialidad
<input type="checkbox"/>	1	1	Maria	Aerobicos
<input type="checkbox"/>	2	2	Pepe	Crossfit
<input type="checkbox"/>	3	3	Juan	Culturismo
*	(NULL)	(NULL)	(NULL)	(NULL)

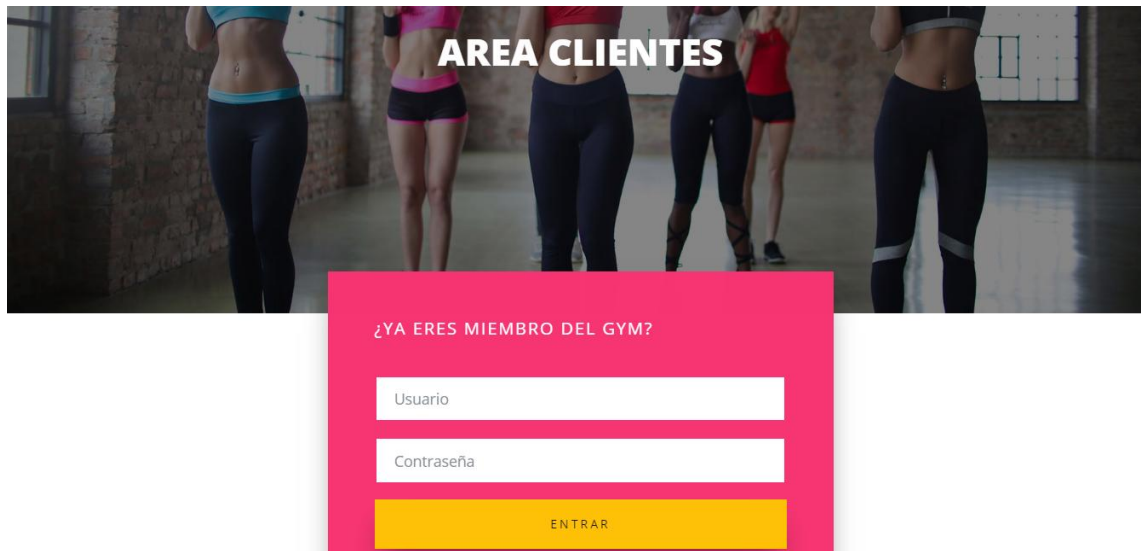


Figura 8 : PANTALLAZO DE AREA DE CLIENTES.

FIGURA 7 Y 8 :

MÓDULO INDEX – MÓDULO LOGIN : En la HOME de la página aparecerá un apartado a la derecha del ÁREA DE CLIENTES , al pulsar se re direcciona a la página de LOGIN .

```
<?php
```

```
    $usuario = $_GET['usuario'];
```

```
    $password = $_GET['password'];
```

```
    echo $usuario;
```

```
    // comprueba en la base de datos tras esta query si existe
```

```
    $query = "SELECT COUNT(*) FROM login WHERE USUARIO =  
    ".$usuario." && PASSWORD= ".$password."";
```

```
    $res = mysqli_query($conexion,$query) or die("debes rellenar los  
    campos");
```

```

/*****

if (isset($_POST['Enviar'])) { // si pulsas boton enviar

    echo "</br>";

    $res = mysqli_query($conexion, $query) or die("debe seleccionar un
    numero");

?>

```

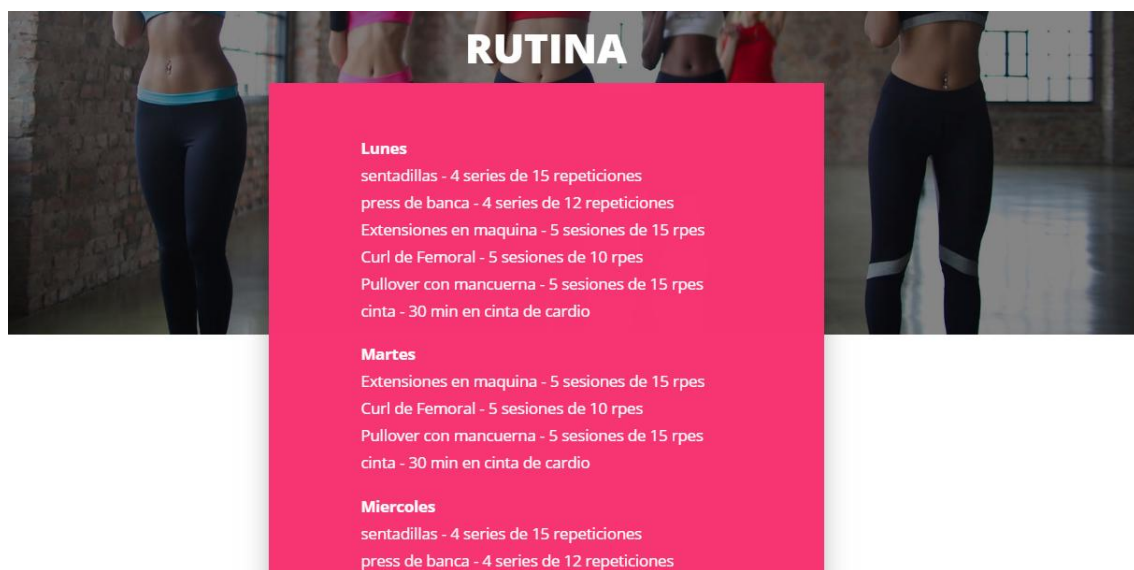


Figura 9: PANTALLAZO DE RUTINA

FIGURA 8 Y 9 :

MÓDULO LOGIN – MÓDULO RUTINA : En la página de LOGIN , podrá acceder el usuario a su área personal de ejercicios o rutinas diarias , siempre y cuando éste , este dado de alta en la base de datos. Al pulsar ENTRAR , enviará la información a módulo RUTINA dónde podrá visualizar todos sus ejercicios.

```
<?php

$usuario = $_POST['usuario'];
$password = $_POST['password'];

echo $usuario;

if(isset ($_POST['enviar'])){

    // comprueba en la base de datos tras esta query si existe
    $query = "SELECT COUNT(*) FROM login WHERE USUARIO =
    ".$usuario." && PASSWORD= ".$password."";

    $res = mysqli_query($conexion,$query) or die("debes rellenar los
    campos");

}

?>
```

ENTRENADORES EXPERTOS

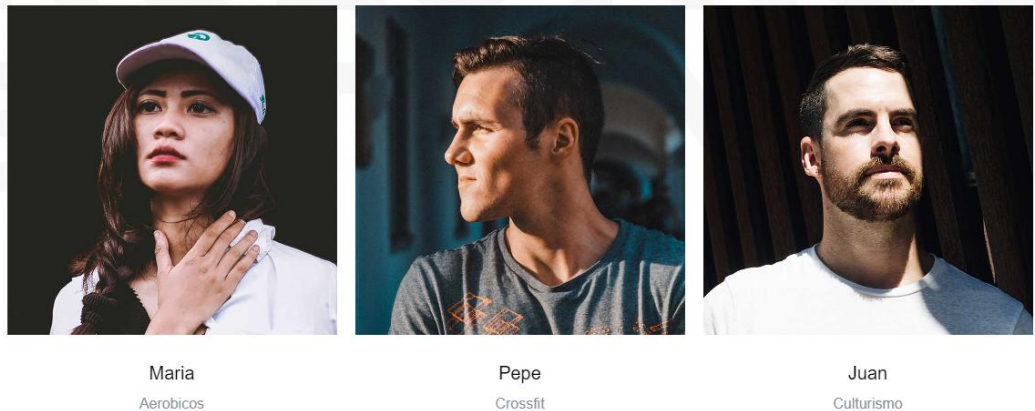


Figura 10: ENTRENADORES

MÓDULO INDEX – ENTRENADORES: En la página de INDEX , el usuario podrá visualizar todos los entrenadores que tiene el gimnasio aparte de las actividades que realiza cada entrenadores , este entrenador esta asociado a un nivel diferente , depende del nivel , tendrá unas actividades dirigidas o otras.

```
<?php
```

```
$query = "SELECT nombre_completo, especialidad FROM entrenadores";
$rs = mysqli_query($conexion, $query) or
die(mysqli_error($conexion));
while($row = mysqli_fetch_array($rs)) {
```

```
?>
```

```
<div>
```

```
<div class="media d-block media-custom text-center">
```

```
<a href="#"></a>
```

```
<div class="media-body">
```

```
<h3 class="mt-0 text-black"><?php echo
```

```
$row['nombre_completo'];?></h3>
```

```
<p class="lead"><?php echo $row['especialidad'];?></p>
```

```
</div>
```

```
</div>
```

```
</div>
```

TEST

	GLOBAL
	MODULAR
	UNITARIO

Los casos de uso , son todos los pasos que deberíamos de hacer cualquier usuario para poder arrancar la aplicación. En la funcionalidad nos aparecerá lo que queremos hacer , por ejemplo LOGIN , en el caso de prueba nos aparecerá todos los posibles casos que puede pasar , en la siguiente columna el Resultado Esperado y en la siguiente el Resultado Obtenido después de probar con la aplicación ese caso de prueba . Los Datos utilizados , si es una prueba de Interfaz o de Base de Datos o ambas , y para finalizar si queremos añadir alguna observación de la prueba importante ha tener en cuenta.

-APLICACIÓN JAVA.

En este apartado nos dedicaremos a probar la aplicación java , que funciona correctamente , con todos sus posible casos de uso , y si realmente se ejecuta uno por uno sin errores , y con todos los posibles casos de prueba que podríamos tener .

A continuación probaremos también la aplicación web PHP , con los mismos pasos y características que daremos en la aplicación JAVA.

Número Caso Prueba	Funcionalidad	Descripción del Caso de Prueba	Resultado Esperado	Datos utilizados	Estado Caso	Observaciones
1	LOGIN	caso 1 : Usuario existente	Valida usuario y permite entrar a plataforma	Base de datos, Interface	OK	
		caso 2 : Usuario que no existe	Mensaje de error y no entra a plataforma		OK	
2	DAR DE ALTA CLIENTE	caso 1 : Usuario existente ya base de datos	Mensaje de error y no entra da de alta .	Base de datos , Interface	OK	
		caso 2 : Falte algún campo por añadir	Mensaje de error , falta algún campo por rellenar	Interface		
		caso 3 : Todos los campos correctos	Dejar dar de alta el cliente nuevo	Base de datos , Interface	OK	
3	ELIMINAR UN CLIENTE	caso 1 : Que se elimine correctamente base de datos	Elimine base de datos	Base de datos	OK	
		caso 2 : Que al eliminar se actualice la tabla	Elimine de la tabla que vemos	Interface	OK	
4	MODIFICAR CLIENTE	caso 1 : Que se modifique correctamente de la base de datos	Modifica en la base de datos correctamente todos	Base de datos	OK	
		caso 2 : Que se modifique en la tabla	que se refresquen los datos en la tabla mostrada	Interface	OK	Si no se muestra porque no se pone ENTER , tienes REFRESC
5	BÚSQUEDA CLIENTE	caso 1 : Que no exista en la base de	No mostrara ningún dato , mostrara tabla vacía.	Base de datos, Interface	OK	

		datos				
		caso 2 : Que aparezca el cliente deseado	Mostrará en la tabla todos los clientes que coinciden	Base de datos, Interface	OK	
6	MOSTRAR CLIENTE	caso 1 : Pinchar mostrar clientes	Aparezca la ventana de los clientes	Interface	OK	
7	SALIR	caso 1 : Pinchar botón salir	Se cierra toda la aplicación	Interface	OK	
8	MAIN	caso 1 : Arranque el Main (la actividad principal)	Aparezca la ventana de login	Interface	OK	

-APLICACIÓN PHP.

	GLOBAL
	MODULAR
	UNITARIO

Número Caso Prueba	Funcionalidad	Descripción del Caso de Prueba	Resultado Esperado	Datos utilizados	Estado Caso
1	ACCESO A AREA CLIENTES	caso 1 : Usuario existente	Valida usuario y permite entrar a plataforma	Base de datos, Interface	OK
		caso 2 : Usuario que no existe	Mensaje de error y no entra a AREA DE CLIENTE		OK
2	MOSTRAR RUTINA	caso 1 : Que se muestre el nivel correspondiente.	Mostrará las rutinas depende del nivel del cliente	Base de datos ,Interface	OK
3	ENTRAR	caso 1 : Al poner usuarios correctamente	Entrará en las rutinas	Interface	OK
4	ACCEDER A PHP	caso 1 : Arrancar el index (índice)	Entrará en la HOME de FITPLUS	Interface	OK
		caso 2 : Si no esta conectado al servidor	Error localhost : no encuentra pagina	Interface	OK

BIBLIOGRAFIA (APA)

ColorLib Templates.(05/2018).ColorLib : ColorLib.cm . Recuperado de :
<https://colorlib.com/wp/template/fitnezz/>

Como buscar e integrar plantilla a una aplicación web(Netbeans).(05/2018).Youtube :
Youtube.com Recuperado de :
www.youtube.com/results?search_query=como+insertar+una+plantilla+en+java

Errores PHP

Mysqli_error.(05/2018).php : php.net . Recuperado de :
<http://php.net/manual/es/mysqli.error.php>

Mysqli_connect.(05/2018).php : php.net . Recuperado de :
<https://secure.php.net/manual/es/function.mysqli-connect.php>

Mysqli_result : data_seek.(05/2018).php : php.net . Recuperado de :
http://php.net/manual/es/mysqli-result.data_seek.php

Mysqli_result : fetch_array.(05/2018).php : php.net . Recuperado de :
<http://php.net/manual/es/mysqli-result.fetch-array.php>

Die.(05/2018).Github : Github.com . Recuperado de :
https://github.com/Chandanmattikalli/PhoneBookFinal/search?q=die&unscoped_q=die

MySQL query shows die error in PHP. (05/2018).Stackoverflow : Stackoverflow.com .
Recuperado de :https://stackoverflow.com/questions/10930415/mysql-query-shows-die-errorinphp?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

Para insertar las rutinas / ejercicios en base de datos :

Rutinas de gimnasios (05/2018).Google : Google.com . Recuperado de :
https://www.google.com/search?source=hp&ei=V4sKW-aaF9HzsAGl9JvoAQ&q=rutinas+de+gimnasio+semanal+filetype%3Axls&oq=rutin&gs_l=psyab.3.1.35i39k112j0i67k113j0l5.1144.1720.0.3797.6.5.0.0.0.111.507.2j3.5.0....0...1c.1.64.psy-ab..1.5.507.0..0i131k1.0.KEp6umaos4w

Plantilla guía para testear todo :

Casos de prueba (05/2018).Google : Google.com . Recuperado de : https://www.google.es/search?ei=JtYKW7uKJ4SWsAHKkaqQCg&q=casos+de+prueba+filetype%3Axls&oq=casos+de+prueba+filetype%3Axls&gs_l=psy-ab.3...4831.5429.0.5680.5.5.0.0.0.0.79.359.5.5.0....0...1c.1.64,psy-ab..0.0.0....0.Btr4bNXqnJ4