



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

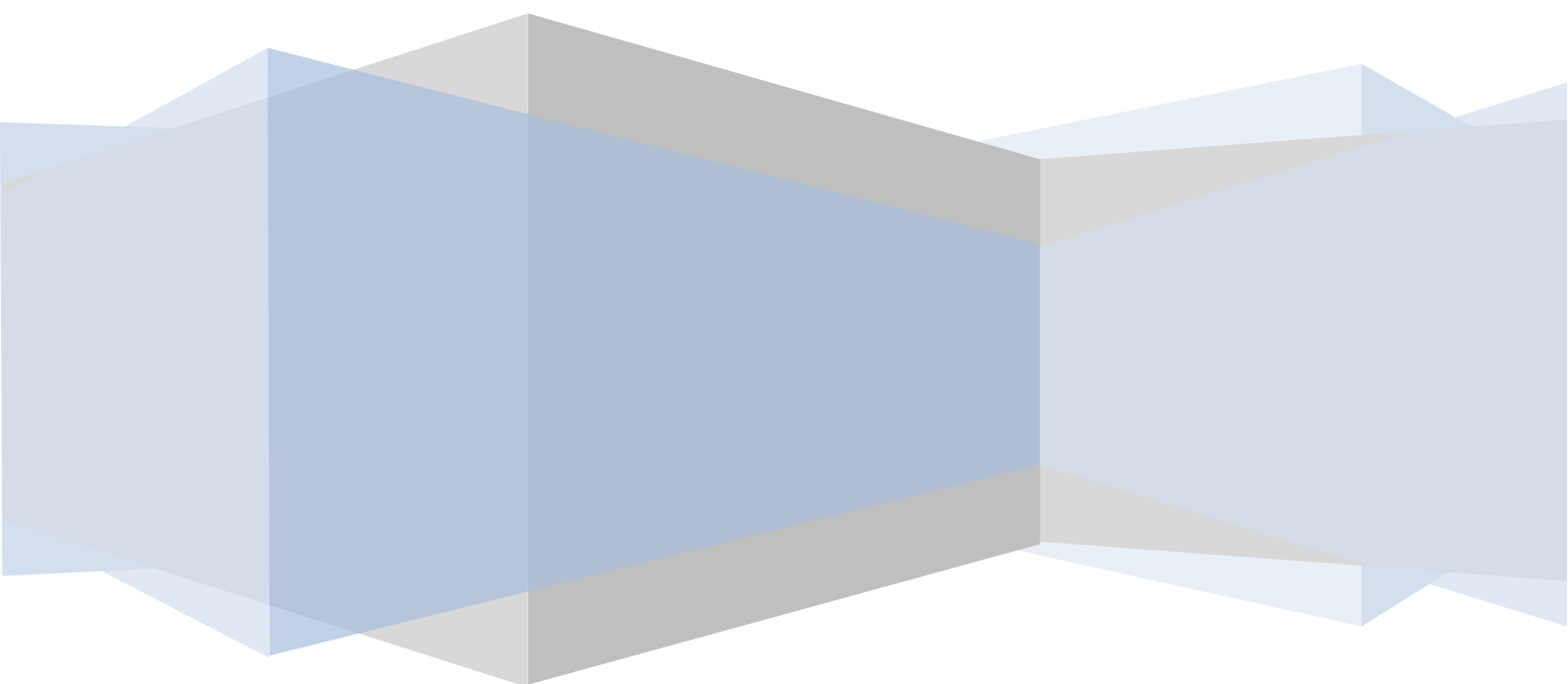


Sistema Computacional para Clubes Deportivos

TEL-332 Multimedia

Francisco Becker

Eduardo Castillo



INDICE

1. INTRODUCCION

1.1 TRASFONDO

1.2 RESUMEN

2. DESCRIPCIÓN GENERAL

2.1 OBJETIVO GENERAL

2.1.1 GENERAL

2.1.2 ESPECÍFICOS

2.1.3 PROBLEMÁTICA QUE ENFRENTA

2.2 DESCRIPCIÓN DE LA SOLUCIÓN

2.3 TECNOLOGÍAS USADAS

3. ESPECIFICACIÓN DE LOS REQUERIMIENTOS

3.1 USUARIOS DEL SISTEMA

3.2 DESCRIPCIÓN DE REQUERIMIENTOS

3.2.1 FUNCIONALES

3.2.2 NO FUNCIONALES

3.2.3 TAREAS DE USUARIO

3.3 FUNCIONES DEL SISTEMA

4. DISEÑO DE LA INTRFAZ DE USUARIO

4.1 ESQUEMA NAVEGACIONAL

4.2 PROTOTIPOS DE PANTALLA

5. DISEÑO DEL SISTEMA

5.1 DISEÑO DE LA ARQUITECTURA

5.2 DISEÑO LÓGICO

5.2.1 DIAGRAMA DE CLASES

5.2.2 DIAGRAMA CASOS DE USO

5.2.3 MODELO DE DATOS

5.2.4 DIAGRAMA DE SECUENCIA

6. IMPLEMENTACIÓN

6.1 DESCRIPCIÓN DE COMPONENTES

7. CONCLUSIONES

1. INTRODUCCIÓN

1.1 TRASFONDO

En la mayoría de los equipos de futbol en Chile poseen un sistema computacional en un estado no actualizado o en un estado ineficiente, el cual dificulta a todos los usuarios, en donde se encuentran tanto los administradores del sistema como los socios del club de futbol, en su uso cotidiano.

Existen grandes deficiencias en estos sistemas, los cuales van desde el tiempo que toma algún usuario para desarrollar alguna tarea, como la falta de alguna funciones que le facilite al usuario en ejecutar tareas en un menor tiempo.

1.2 RESUMEN

El proyecto consiste en el diseño y en la programación de un sistema computacional, tanto para administradores del sistema como socios del club deportivo, el cual solucione las falencias que existen en los actuales sistemas. La idea principal es que los usuarios ejecuten las tareas de una forma más rápida y sencilla.

Cabe mencionar que este proyecto se está desarrollando actualmente para el Club Deportivo Coquimbo Unido, el cual desea actualizar totalmente su sistema por uno que solucione todas sus necesidades.

2. DESCRIPCIÓN GENERAL

2.1 OBJETIVOS

2.1.1 GENERAL

El objetivo principal del desarrollo del proyecto es la de crear un sistema computacional totalmente actualizado para los clubes deportivos, específicamente de fútbol, el cual pueda cumplir con todas las exigencias de los usuarios del sistema.

2.1.2 ESPECÍFICOS

- Se espera que el nuevo sistema computacional impacte de una forma positiva en los usuarios.
- Se intentará cumplir con todas las demandas requeridas por los usuarios.
- Mejorar el tiempo en que los usuarios desarrollen alguna tarea en el sistema.
- Como se manejara una mayor información, se dará importancia a la seguridad a los datos de los usuarios.

2.1.3 PROBLEMÁTICAS QUE ENFRENTA

La problemática que enfrenta el proyecto es el inadecuado sistema computacional existente en la mayoría de los clubes deportivos de fútbol, siendo incapaces de satisfacer todas las necesidades actuales presentes, tanto para los administradores del sistema, como para los socios del club.

2.2 DESCRIPCIÓN DE LA SOLUCIÓN

En este proyecto se realizara la creación de un sistema computacional totalmente actualizado, que satisfaga las actuales necesidades de los usuarios. Este sistema podrá ser utilizado para cualquier club deportivo de fútbol, ya que su principal característica que será un sistema genérico.

Este proyecto consta de dos partes, la primera consiste en el desarrollo de la parte de los administradores del sistema, los cuales tienen diferentes niveles de gestión en el sistema, los de más alto nivel pueden generar reportes, hacer descuentos a personas y acceder completamente a la base de datos, los de nivel intermedio como tarea principal solo pueden generar reportes, y los de bajo nivel pueden ingresar socios, ver los estados de cuentas de estos, etc. La segunda parte del proyecto consiste en el desarrollo de la parte de los socios del club, los cuales como principales tareas son ver la disponibilidad de asientos en el estadio, tarifas de los partidos, etc.

2.3 TECNOLOGÍAS USADAS

Las tecnologías que se han utilizado para el desarrollo del proyecto son las siguientes:

- Base de datos MySQL
- Lenguajes de programación PHP y JavaScript por parte del servidor.
- Computador para el servidor.
- Entorno de desarrollo DreamWeaver CS5.
- Hojas de Estilo para el manejo de la estructura de la página.

Se analizará posteriormente si se desarrollara una aplicación móvil, dependiendo principalmente del tiempo, la cual estará enfocada a los socios del club, los cuales podrán ejecutar las mismas tareas anteriormente expuestas.

3. ESPECIFICACIÓN DE LOS REQUERIMIENTOS

3.1 USUARIOS DEL SISTEMA

Los usuarios del sistema computacional son los siguientes:

Administradores: Estos son los principales usuarios del sistema, los cuales están divididos en diferentes niveles dependiendo de las responsabilidades que posean.

Socios: Estos deberán estar registrados en la base de datos para poder utilizar todas las tareas.

3.2 DESCRIPCIÓN DE REQUERIMIENTOS

3.2.1 FUNCIONALES

Requerimiento-1

Implementación de un Login en el servidor, que le permita a los usuarios ingresar al sistema con solo los permisos que este posea.

Requerimiento-2

Implementación de un formulario para el registro de los socios del club.

Requerimiento-3

Se deberá tener la opción de editar la información de los socios.

Requerimiento-4

Despliegue de los diferentes interfaces para los administradores del sistema y poder ejecutar las tareas que pueden ejercer.

Requerimiento-5

Servidor deberá saber la congestión del sistema.

Requerimiento-6

El socio debe poder desplegar todos las opciones y poder ejecutar cualquier tarea.

Requerimiento-7

El servidor deberá ser capaz de generar diferentes estadísticas de la información de los usuarios.

Requerimiento-8

El sistema deberá entregar un reporte de datos de interés para los administradores.

3.2.2 NO FUNCIONALES

A continuación se especifican los requerimientos no funcionales del proyecto:

COMPATIBLE

Se creara un sistema computacional que podrá ser utilizado por cualquier club deportivo.

SEGURIDAD

Se debe de proveer la seguridad necesaria para que todos los datos de los usuarios estén alojados de una forma segura.

USABILIDAD

El sistema debe de desarrollada de una forma en que los diferentes usuarios puedan desenvolverse de una forma rápida y sencilla, que no posea barreras que interfieran en las tareas que desarrollan.

ESCALABLE

El sistema debe ser desarrollado para el caso en que en el futuro se generen nuevas necesidades, se puedan agregar las funciones necesarias para la solución de estos.

3.2.3 TAREAS DE USUARIO

A continuación se especificaran las tareas de los diferentes usuarios del sistema computacional:

Administradores: Las tareas que cumplen dependen del nivel que posean estos, los cuales son los siguientes:

Nivel alto: Las tareas principales que cumplen se encuentran la generación de reportes, establecer los descuentos para los socios, además de poder ingresar completamente a la información de la base de datos, creación de planes, encardados de la Black List, registro de los socios.

Nivel medio: Las tareas principales que cumplen se encuentran la generación de reportes, creación de eventos, registro de socios y las cuentas de los socios.

Nivel bajo: Las tareas principales que cumplen se encuentran el ingreso de socios al club, ver el estado de las cuentas de los socios.

Socios: Para poder ser socios deben de estar registrados en la base de datos del club, el cual les da derecho a todos los beneficios que se ofrecen. Entre las cuales son de ver la disponibilidad de asientos en el estadio, ver las tarifas preferenciales a las que puede optar, como también a promociones que puede acceder.

3.3 FUNCIONES DEL SISTEMA

El sistema va a cumplir las siguientes funciones:

POR PARTE DEL SERVIDOR:

- El servidor debe de controlar las diferentes sesiones de los usuarios del sistema, esto es desplegar las diferentes interfaces para los diferentes niveles de administradores, como la de la interfaz de socio.
- El servidor debe de poder registrar todas las modificaciones que los usuarios generen en el sistema, estos serán el ingreso de nueva información, como también la modificación y eliminación.
- En situaciones en que el sistema ocurra un error, deberá ser capaz de informar a los usuarios de este estado.
- El servidor deberá de desplegar todos los datos requeridos por los usuarios.
- El servidor deberá de asignar un id dinámicamente a los usuarios al ingresar al sistema, para poder registrar todos las tareas realizadas por este.

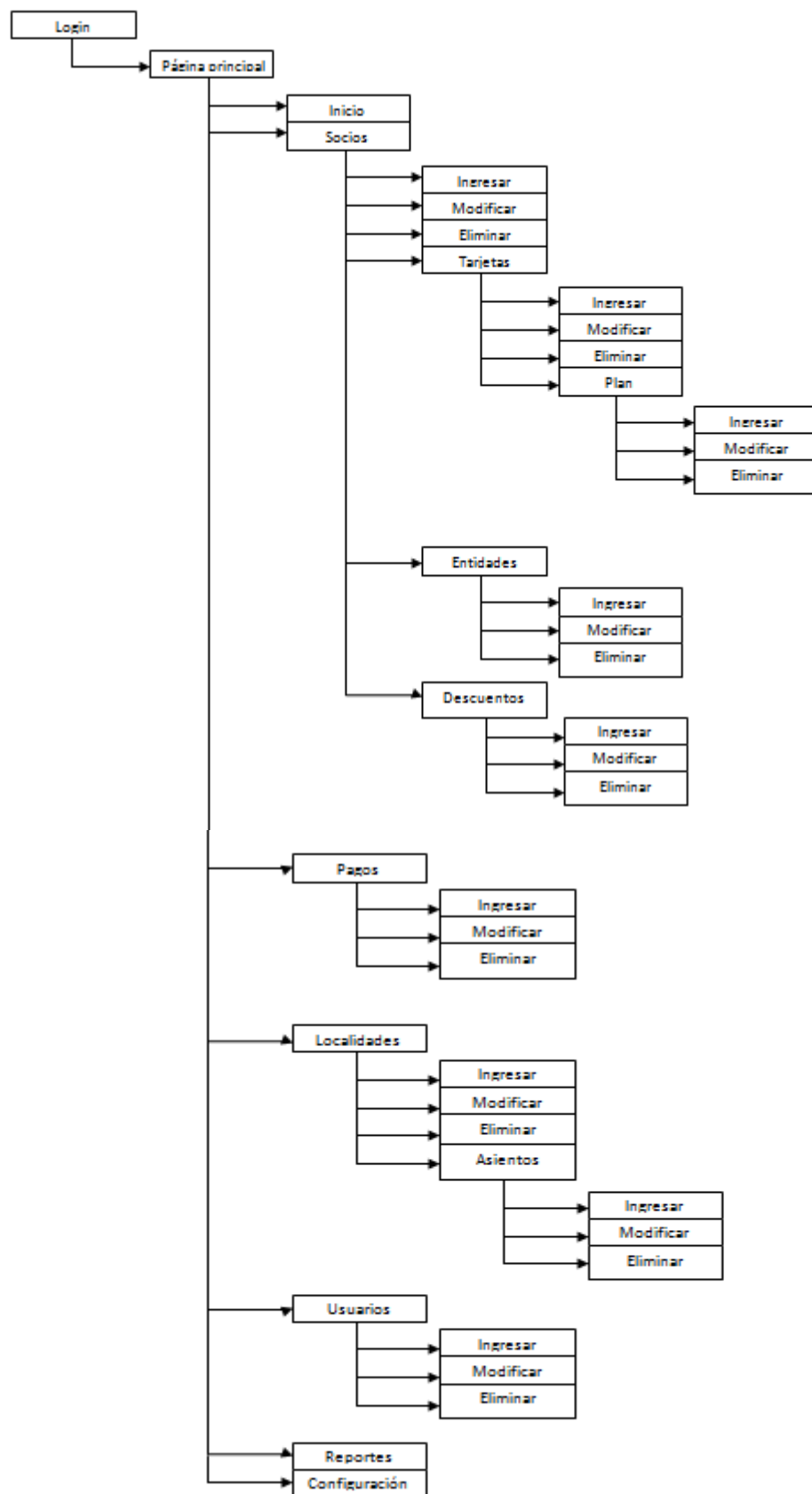
POR PARTE DE LOS USUARIOS:

- El usuario debe ser capaz de ingresar los diferentes datos de las tareas asignadas.
- El usuario debe poder desplegar las diferentes opciones en el sistema a las cuales puede ingresar.
- El usuario debe ser capaz de modificar los diferentes datos de las tareas asignadas.
- El usuario debe ser capaz de eliminar los diferentes datos de las tareas asignadas.

4. DISEÑO DE LA INTERFAZ DE USUARIO

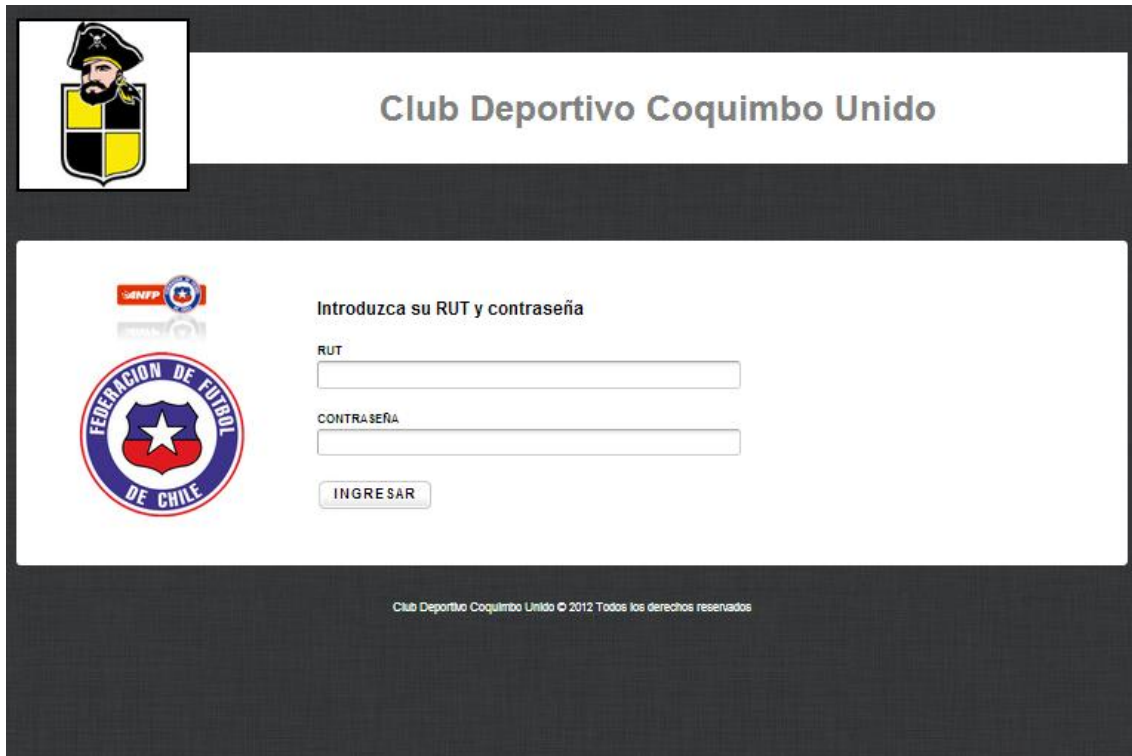
4.1 ESQUEMA NAVEGACIONAL

A continuación se mostrara el esquema navegacional solamente de los administradores del sistema.



ESQUEMA NAVEGACIONAL

4.2 PROTOTIPOS DE PANTALLA



The login screen features a dark grey header with the club's logo on the left and the name "Club Deportivo Coquimbo Unido" in the center. Below the header, on the left, is a circular logo for the "FEDERACION DE FUTBOL DE CHILE" with a star in the center. To the right of this logo is a login form with the title "Introduzca su RUT y contraseña". The form contains two input fields: "RUT" and "CONTRASEÑA", followed by an "INGRESAR" button. At the bottom of the page, a small copyright notice reads "Club Deportivo Coquimbo Unido © 2012 Todos los derechos reservados".

Figura 4.2.1 Login servidor



The main page has a dark grey header with the club's logo and name. Below the header is a navigation menu with the following items: "INICIO" (highlighted), "SOCIOS", "PAGOS", "LOCALIDADES", "USUARIOS", "REPORTES", and "CONFIGURACIÓN". The main content area is white and contains a user profile section on the left with the following details: "NOMBRE", "Juan", "Castillo", "RUT", and "123". To the right of this section is the text "Contenido inicial". At the bottom of the profile section is a "CERRAR SESIÓN" button. A small copyright notice at the bottom of the page reads "Club Deportivo Coquimbo Unido © 2012 Todos los derechos reservados" followed by "jQuery Menu by Apycom".

Figura 4.2.2 Página principal



Club Deportivo Coquimbo Unido

INICIO SOCIOS PAGOS LOCALIDADES USUARIOS REPORTES CONFIGURACIÓN

NOMBRE

Juan

Castillo

RUT

123

[CERRAR SESIÓN](#)

Ingresar Usuario

NOMBRES

APELLIDOS

RUT

CONTRASEÑA

NIVEL DE ACCESO

[Elegir una opción](#)

[INGRESAR](#)

Club Deportivo Coquimbo Unido © 2012 Todos los derechos reservados | Query Menu by Aqycom

Figura 4.2.3 Ingresar Socio



Club Deportivo Coquimbo Unido

INICIO SOCIOS PAGOS LOCALIDADES USUARIOS REPORTES CONFIGURACIÓN

NOMBRE

Juan

Castillo

RUT

123

[CERRAR SESIÓN](#)

Ingresar Entidad

NOMBRE

FONO

DIRECCION

EMAIL

TIPO DE ENTIDAD

SOCIO TITULAR

[INGRESAR](#)

Club Deportivo Coquimbo Unido © 2012 Todos los derechos reservados | Query Menu by Aqycom

Figura 4.2.4 Ingresar Entidad



Club Deportivo Coquimbo Unido

INICIO SOCIOS PAGOS LOCALIDADES USUARIOS REPORTES CONFIGURACIÓN

NOMBRE

Juan

Castillo

RUT

123

CERRAR SESIÓN

Ingresar Localidad

LOCALIDADES EXISTENTES

No existen localidades almacenadas

NOMBRE

INGRESAR

Club Deportivo Coquimbo Unido © 2012 Todos los derechos reservados jQuery Menu by Apycom

Figura 4.2.5 Ingresar Localidad



Club Deportivo Coquimbo Unido

INICIO SOCIOS PAGOS LOCALIDADES USUARIOS REPORTES CONFIGURACIÓN

NOMBRE

Juan

Castillo

RUT

123

CERRAR SESIÓN

Ingresar Plan

Galeria ☐

Marquesina ☐

Tribuna Alta ☐

Tribuna Baja ☐

Vip ☐

Particular ☐

Familiar ☐

VALOR

\$

FECHA

INGRESAR

Club Deportivo Coquimbo Unido © 2012 Todos los derechos reservados jQuery Menu by Apycom

Figura 4.2.6 Ingresar Plan

5. DISEÑO DEL SISTEMA

5.1 DISEÑO DE LA ARQUITECTURA

El diseño se especifica en la siguiente figura:



Diseño de Arquitectura

Se utilizó el Modelo de Tres Capas para el diseño de la arquitectura, la cual consiste en:

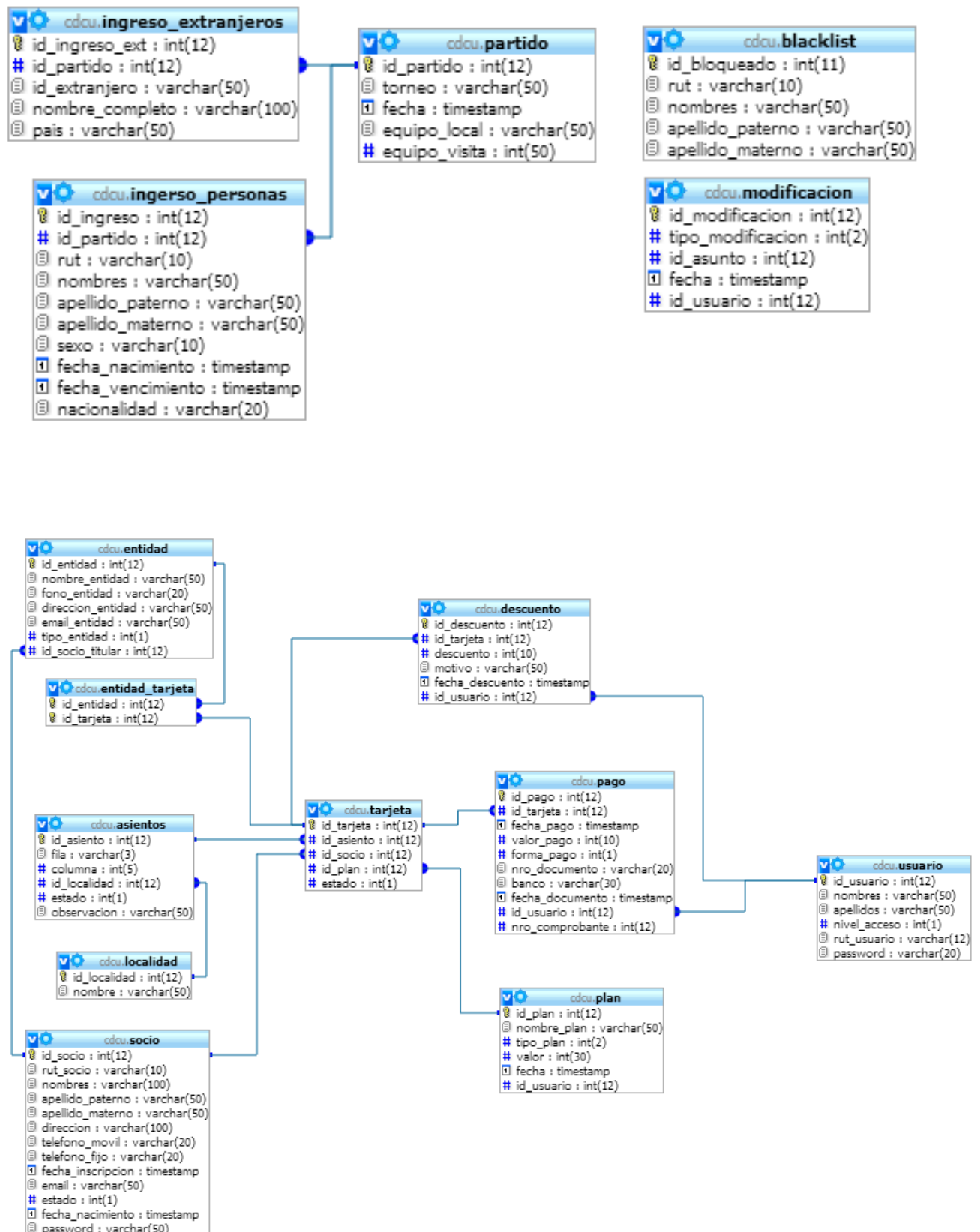
- Interfaz
- Capa de Negocios
- Base de Datos

El modelo fue elegido ya que las capas se encuentran desligadas, lo que significa que al realizar algún cambio solo se aplica a la capa correspondiente, lo cual facilita el trabajo de realizar los cambios.

5.2 DISEÑO LÓGICO

5.2.1 DIAGRAMA DE CLASES

A continuación se muestra el diagrama de clases que se desarrolló para el proyecto:



Diseño de Diagrama de Clases

Cabe mencionar que el modelo puede ser modificado en el desarrollo del proyecto.

5.2.2 DIAGRAMA CASOS DE USO

El diagrama de casos de uso representa a los diferentes actores, Administradores y Socios, detallando las funcionalidades a las que pueden acceder.

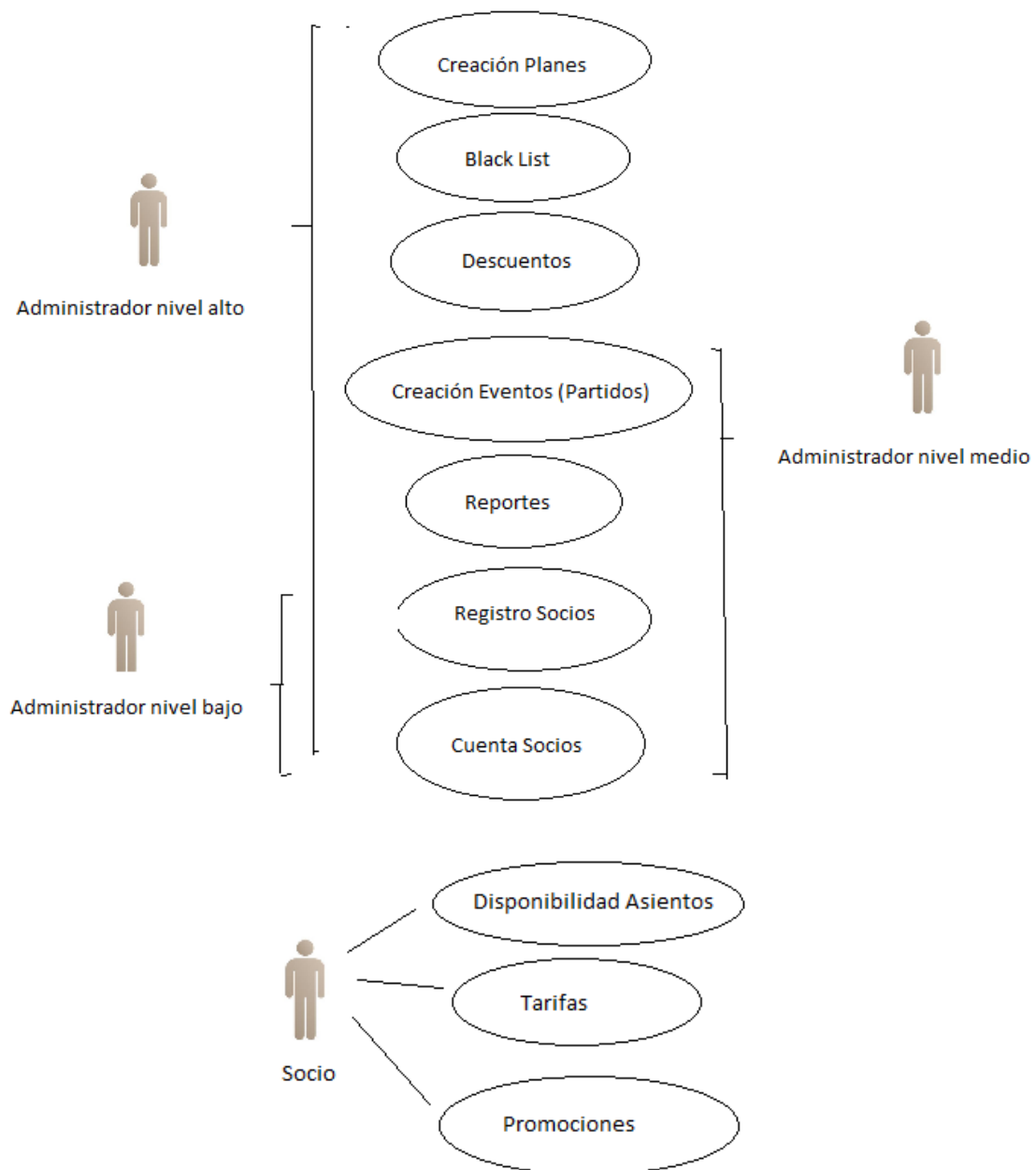
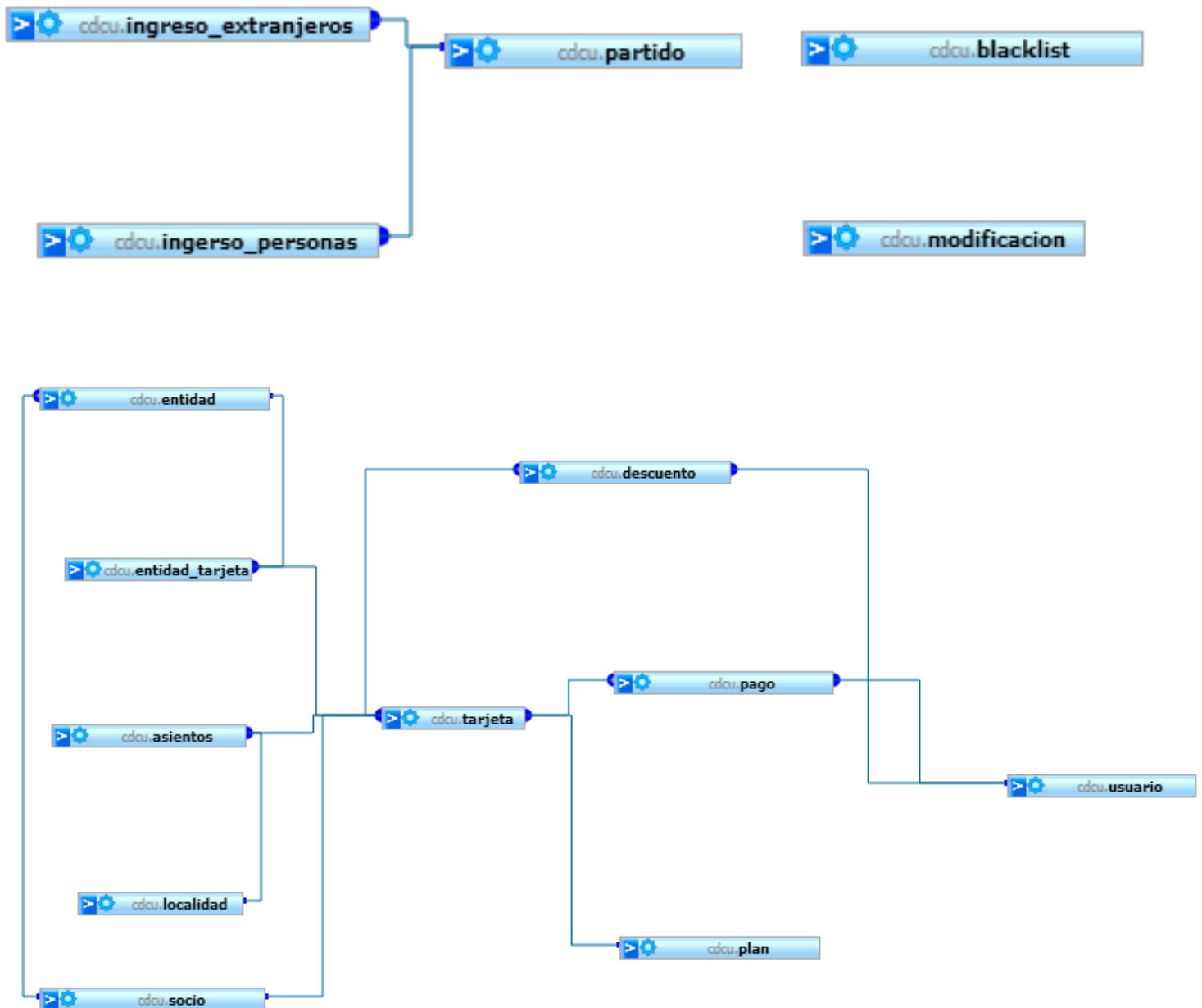


Diagrama Casos de Uso

5.2.3 MODELO DE DATOS

A continuación se muestra el Modelo de Datos que se desarrollo para el proyecto, mostrando las diferentes claves primarias y foráneas de cada una de las tablas.

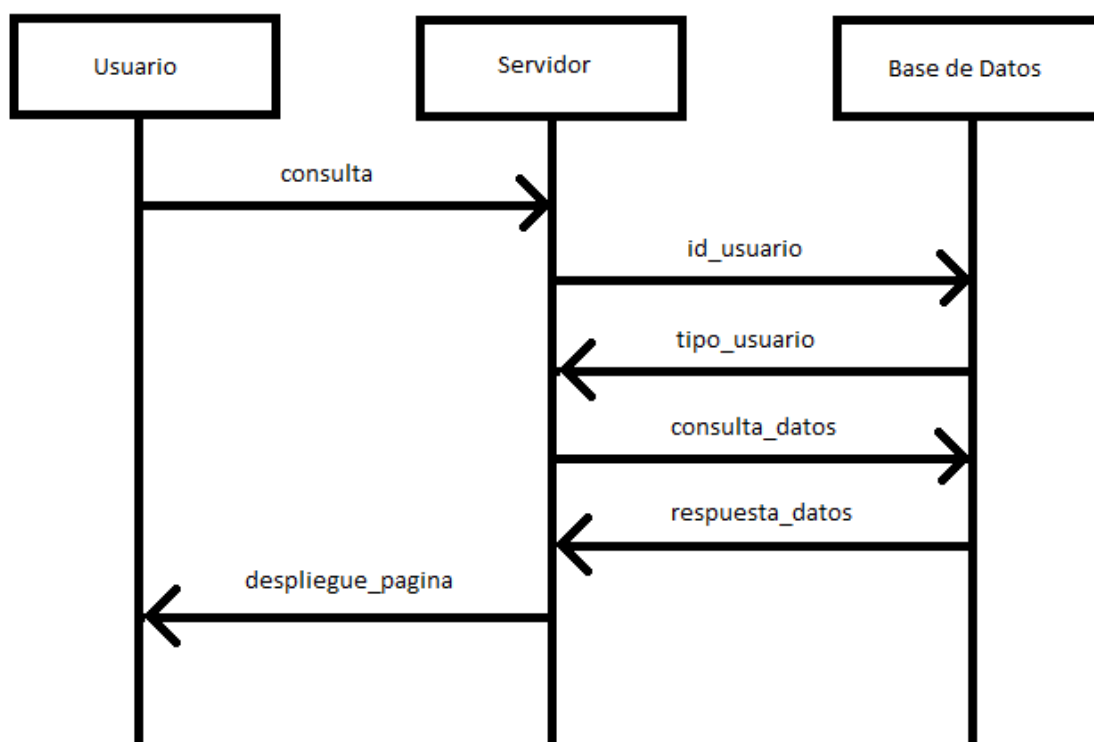


Diseño de Modelo de Datos

5.2.4 DIAGRAMA DE SECUENCIA

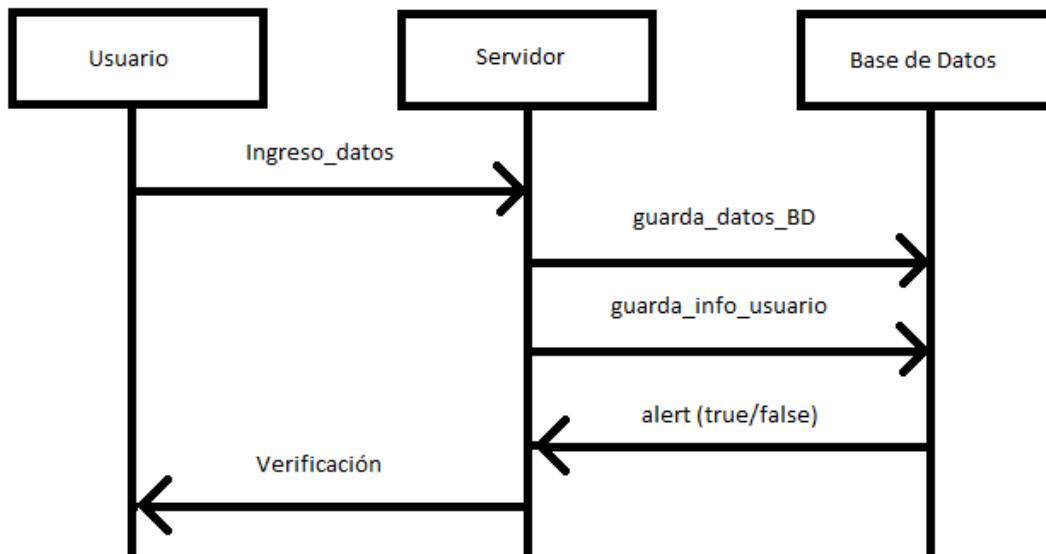
A continuación se muestra los Diagramas de Secuencias existentes en el sistema actualmente, las cuales pertenecen solamente a la de los administradores:

El siguiente diagrama de secuencia generaliza el inicio de sesión de los administradores, los cuales tienen ciertos permisos en sistema dependiendo del nivel que posean.



Esquema Despliegue de interfaz de usuario

El siguiente diagrama de secuencia generaliza la modificación de la información en la base de datos, los cuales pueden ser el ingreso de nueva información, como la modificación y eliminación de información existente.



Esquema Ingreso de información

6. IMPLEMENTACIÓN

6.1 DESCRIPCIÓN DE COMPONENTES

A continuación se expondrá el código de importancia actualmente desarrollados en el proyecto, los cuales son solamente por parte de las tareas de los administradores, los cuales son los siguientes.

index.php

```

<?php
    session_start();
    if(isset($_SESSION["id"]))
        header("Location: home.php");
?>
  
```

Consiste en el sistema de ingreso del sistema, el cual el usuario deberá ingresar su R.U.T y contraseña para poder ser identificado y desplegar la página con solo los permisos permitidos, el cual se muestra a continuación.

home.php

```
<?php
    session_start();
    $id = 0;
    if(isset($_SESSION["id"]))
        $id = $_SESSION["id"];
    if(!$id)
        header("Location: index.php");
?>

$.ajax({
    type: "POST",
    url: "datosUsuario.php",
    data: data2,
    success: function(result)
    {
        $("#rut_usuario").html(result);
    }
});
```

doLogin.php

Consiste en el código necesario el cual le asigna un id a los usuarios cuando inician la sesión, esto es para poder registrar todas las acciones de los usuarios en el sistema. El código se muestra a continuación.

```
<?php

    session_start();
    $id = 0;
    if(isset($_SESSION["id"]))
        $id = $_SESSION["id"];
    if(!$id)
        header("Location: index.php");

    include 'includes/header.php';

    if(isset($_POST["username"]) && isset($_POST["password"])) {
        $u = new Usuario();
        $u->setRutUsuario($_POST["username"]);
        $u->setPassword($_POST["password"]);
    }
```

```

        if($u->login())
        {
            session_register("id");
            $_SESSION["id"] = $u->getId();
            unset($u);
            echo "success";
        }
        else
        {
            session_destroy();
            unset($u);
            echo "error";
        }
    }
}

?>

```

doLogout.php

Consiste en el código necesario para el cierre de sesión del usuario. El código se muestra a continuación.

```

<?php

session_start();
$id = 0;
if(isset($_SESSION["id"]))
    $id = $_SESSION["id"];
if(!$id)
    header("Location: index.php");

session_destroy();
echo "success";

?>

```

ingresarEntidad.php - ingresaEntidad.html

Consiste en el código necesario para la creación de las diferentes entidades que pueden existir. El código se muestra a continuación.

```
<?php
    session_start();
    $id = 0;
    if(isset($_SESSION["id"]))
        $id = $_SESSION["id"];
    if(!$id)
        header("Location: index.php");

    include 'includes/header.php';

    $nombre = $_POST['nombre'];
    $fono = $_POST['fono'];
    $direccion = $_POST['direccion'];
    $email = $_POST['email'];
    $tipo = $_POST['tipo'];
    $titular = $_POST['titular'];
    $entidad = new Localidad('', $nombre, $fono, $direccion, $email, $tipo, $titular);

    if($entidad->save())
        echo "success";
    else
        echo "error";

?>

$.ajax({
    type: "POST",
    url: action,
    data: form_data,
    success: function(response)
    {
        if(response == "success") {
            alert("Entidad ingresada correctamente");
            $("#form_ing_ent").each (function() {
                this.reset();
            });
        }
        else
            alert("Error al ingresar localidad");
    }
});
```

ingresarLocalidad.php - ingresarLocalidad.html

Consiste en el código necesario para la creación de las diferentes Localidades que pueden existir en el interior del estadio. El código se muestra a continuación.

```
<?php
```

```
    session_start();
    $id = 0;
    if(isset($_SESSION["id"]))
        $id = $_SESSION["id"];
    if(!$id)
        header("Location: index.php");

    include 'includes/header.php';

    $nombre = $_POST['nombre'];
    $localidad = new Localidad('', $nombre);
    if($localidad->save())
        echo "success";
    else
        echo "error";
```

```
?>
```

```
$.ajax({
    type: "POST",
    url: action,
    data: form_data,
    success: function(response)
    {
        if(response == "success") {
            alert("Localidad ingresada correctamente");
            $("#localidades").append("<p>" + $("#nombre").val() + "</p>");
            $("#form_ing_loc").each (function() {
                this.reset();
            });
        }
        else
            alert("Error al ingresar localidad");
    }
});
```


ingresarPlan.php - ingresarPlan.html

Consiste en el código necesario para la creación de los diferentes planes que pueden existir para los socios. El código se muestra a continuación.

```
<?php
    session_start();
    $id = 0;
    if(isset($_SESSION["id"]))
        $id = $_SESSION["id"];
    if(!$id)
        header("Location: index.php");

    include 'includes/header.php';

    $nombre_localidad = $_POST['nombre'];
    $agrupacion = $_POST['agrupacion'];
    $valor = $_POST['valor'];
    $fecha = $_POST['fecha'];
    $hora = date("h:i:s");

    $plan = new Plan('', $nombre_localidad." ".$agrupacion, 2, $valor, $fecha." ".$hora, $id);
    if($plan->save())
        echo "success";
    else
        echo "error";

?>
```

```
$.ajax({
    type: "POST",
    url: action,
    data: form_data,
    success: function(response)
    {
        if(response == "success") {
            alert("Plan ingresado correctamente");
            $("#form_ing_plan").each (function() {
                this.reset();
            });
            $('input[name="agrupacion"]').attr('checked', false);
        }
        else
            alert("Error al ingresar plan");
    }
});
```

ingresarSocio.php - ingresarSocio.html

Consiste en el código necesario para el ingreso de los socios del club deportivo. El código se muestra a continuación.

```
<?php
```

```
    session_start();
    $id = 0;
    if(isset($_SESSION["id"]))
        $id = $_SESSION["id"];
    if(!$id)
        header("Location: index.php");

    include 'includes/header.php';

    $rut = $_POST["rut"];
    $nombres = $_POST["nombres"];
    $apellido_paterno = $_POST["apellido_paterno"];
    $apellido_materno = $_POST["apellido_materno"];
    $direccion = $_POST["direccion"];
    $telefono_movil = $_POST["celular"];
    $telefono_fijo = $_POST["telefono_fijo"];
    $fecha_inscripcion = $_POST["fecha_inscripcion"];
    $email = $_POST["email"];
    $estado = 1;
    $fecha_nacimiento = $_POST["fecha_nacimiento"];
    $password = $_POST["password"];

    $s = new Socio('', $rut, $nombres, $apellido_paterno,
    $apellido_materno, $direccion, $telefono_movil, $telefono_fijo,
    $fecha_inscripcion, $email, $estado, $fecha_nacimiento, $password);
    if($s->save())
        echo "success";
    else
        echo "error";
```

```
?>
```

```

$.ajax({
    type: "POST",
    url: action,
    data: form_data,
    success: function(response)
    {
        if(response == 'success') {
            alert('Socio ingresado correctamente');
            $("#form_ing_socio").each (function() {
                this.reset();
            });
        }
        else
            alert('Error al ingresar socio');
    }
});

```

ingresarUsuario.php - ingresarUsuatio.html

Consiste en el código necesario para la creación de diferentes administradores del sistema, el cual le proporciona los permisos necesarios para cumplir con sus tareas. El código se muestra a continuación.

<?php

```

session_start();
$id = 0;
if(isset($_SESSION["id"]))
    $id = $_SESSION["id"];
if(!$id)
    header("Location: index.php");

include 'includes/header.php';

$is_ajax = $_REQUEST['is_ajax'];
if(isset($is_ajax) && $is_ajax)
{
    $nombres = $_REQUEST['nombres'];
    $apellidos = $_REQUEST['apellidos'];
    $rut = $_REQUEST['rut'];
    $password = $_REQUEST['password'];
    $nivel_acceso = $_REQUEST['nivel_acceso'];
}

```

```

        $usuario = new Usuario('', $nombres, $apellidos,
        $nivel_acceso, $rut, $password);

        if($usuario->save())
        {
            echo "success";
        }
    }

?>

$.ajax({
    type: "POST",
    url: action,
    data: form_data,
    success: function(response)
    {
        if(response == 'success')
            alert('Usuario ingresado correctamente');
        else
            alert('Error al ingresar usuario');
    }
});

```

A continuación se mostrara algunos códigos, por asuntos de espacio, de las clases existentes en el sistema.

Clase de la Base de Datos:

```

<?php

class BaseDatos
{

    function BaseDatos() {
    }

    function connectClass() {
        global $strings, $oConn, $databaseType;
        $oConn= @mysql_connect(MYSERVER, MYLOGIN, MYPASSWORD) or die($strings["error_server"]);
        mysql_select_db(MYDATABASE, $oConn) or die($strings["error_server"]);
    }

    function query($sql) {
        global $oConn;
        $this->$sql = $sql;
        mysql_query("SET NAMES 'utf8'", $oConn);
        $this->result = @mysql_query($sql, $oConn);
        return $this->result;
    }
}

```

```

function fetch() {
    global $row;
    if (!$this->result) {
        echo "Un Error ha ocurrido.\n";
        echo "<font color=red><b>".mysql_error($row)."</b></font><br><br><br>";
        echo "<font color=red><b>La consulta es: \"".$this->sql."</b></font><br>";
        exit;
    }
    $row=@mysql_fetch_array($this->result);
    return $row;
}

function close() {
    global $oConn;
    @mysql_free_result($this->result);
    @mysql_close($oConn);
}

function insertbd($query) {
    global $oConn, $row, $sql;
    $this->connectClass();
    $sql = $query;
    $result = $this->query($sql);
    $this->close();

function insertbd($query) {
    global $oConn, $row, $sql;
    $this->connectClass();
    $sql = $query;
    $result = $this->query($sql);
    $this->close();

    if ($result) {
        return true;
    } else {
        global $msg;
        $msg = $sql;
        return false;
    }
}

function updatebd($sqlquery) {
    global $oConn, $row;
    $this->connectClass();
    $sql = $sqlquery;
    $result = $this->query($sql);
    $this->close();

```

```

        if ($result) {
            return true;
        } else {
            global $msg;
            $msg = $sql;
            return false;
        }
    }

function updatebd($sqlquery) {
    global $oConn, $row;
    $this->connectClass();
    $sql = $sqlquery;
    $result = $this->query($sql);
    $this->close();
    |
    if($result) {
        return true;
    } else {
        return false;
    }
}

```

```

function deletebd($sqlquery) {
    return $this->insertbd($sqlquery);
}

}

?>

```

Clase de Usuario:

```
<?php

require_once("UsuarioBD.class.php");

class Usuario extends UsuarioBD {

    function Usuario($id_usuario="", $nombres="", $apellidos="",
    $nivel_acceso="", $rut_usuario="", $password="") {
        $this->id_usuario = $id_usuario;
        $this->nombres = $nombres;
        $this->apellidos = $apellidos;
        $this->nivel_acceso = $nivel_acceso;
        $this->rut_usuario = $rut_usuario;
        $this->password = $password;
    }

    function setId($id_usuario) {
        $this->id_usuario = $id_usuario;
    }

    function setNombres($nombres) {
        $this->nombres = $nombres;
    }
}
```

```

function setApellidos($apellidos) {
    $this->apellidos = $apellidos;
}

function setNivelAcceso($nivel_acceso) {
    $this->nivel_acceso = $nivel_acceso;
}

function setRutUsuario($rut_usuario) {
    $this->rut_usuario = $rut_usuario;
}

function setPassword($password) {
    $this->password = $password;
}

function getId() {
    return $this->id_usuario;
}

function getNombres() {
    return $this->nombres;
}

function getApellidos() {
    return $this->apellidos;
}

function getNivelAcceso() {
    return $this->nivel_acceso;
}

function getRutUsuario() {
    return $this->rut_usuario;
}

function getPassword() {
    return $this->password;
}

}

?>

```


7. CONCLUSIONES

En el presente avance del proyecto se pudo concluir lo siguiente:

- No se ha alcanzado el 50% de desarrollo del proyecto, ya que se han presentado diferentes dificultades en el transcurso de este.
- Como se ha desarrollado con el Modelo Tres Capas, es fácilmente proceder a algún cambio en el código.
- Como es un proyecto que se está actualmente desarrollando para un club deportivo, es probable que se hagan modificaciones futuras.
- En el informe final se entregara con el desarrollo final del proyecto, ya realizadas las modificaciones.
- Hasta el momento se han respondido a los diferentes requerimientos que ha establecido el cliente.