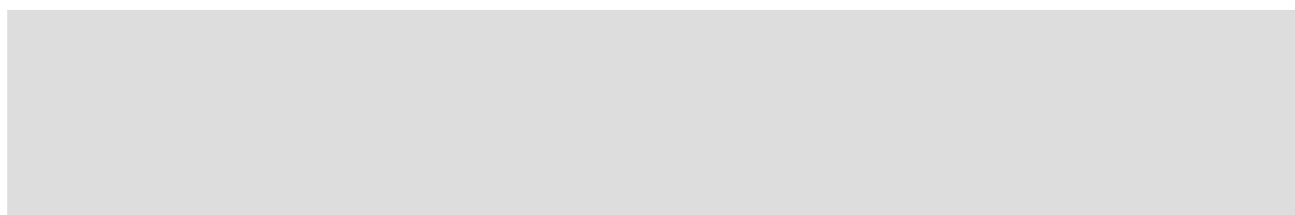


Cliente: Luney Pilates

Modulo:



***CATEDRA DE ING.  
DE SOFTWARE***



# Ingeniería en Software I

Trabajo Práctico Número:

Grupo Número:

Integrantes: Gomez Veliz Kevin Shionen  
Quiroga Gonzalo

Fecha de Entrega 1:

Estado:

Fecha de Entrega 2:

Estado:

Fecha de Entrega 3:

Estado:

Observaciones:

Entrega 1:

Entrega 2:

Entrega 3:

## **I. Introducción**

Esta especificación tiene como objetivo analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar. Para ello, se identificarán los requisitos que ha de satisfacer el nuevo sistema mediante entrevistas, el estudio de los problemas de las unidades afectadas y sus necesidades actuales. Además de identificar los requisitos se deberán establecer prioridades, lo cual proporciona un punto de referencia para validar el sistema final que compruebe que se ajusta a las necesidades del usuario.

## **II. Identificación de usuarios participantes**

Los objetivos de esta tarea son identificar a los responsables de cada una de las unidades y a los principales usuarios implicados. En la organización se identificaron los siguientes usuarios:

- **Propietario del Gimnasio:** propietario y administrador del gimnasio.
- **Grupo de entrenadores del gimnasio:** Formado por los entrenadores.

Es de destacar la necesidad de una participación activa de los usuarios del futuro sistema en las actividades de desarrollo del mismo, con objeto de conseguir la máxima adecuación del sistema a sus necesidades y facilitar el conocimiento paulatino, permitiendo una rápida implantación.

## **III. Catálogo de Requisitos del Sistema**

El objetivo de la especificación es definir en forma clara, precisa, completa y verificable todas las funcionalidades y restricciones del sistema que se desea construir. Esta documentación está sujeta a revisiones por el grupo de usuarios que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación por parte de la dirección de 'Luney Pilates' y del grupo de usuarios. Una vez aprobado, servirá de base al equipo para la construcción del nuevo sistema.

Esta especificación se ha realizado de acuerdo al estándar "IEEE Recommended Practice for Software Requirements Specifications (IEEE/ANSI 830-1993)", y se basa en las entrevistas realizadas a los usuarios participantes y el estudio de la documentación existente.

### **III.1. Objetivos y Alcance del Sistema**

Los principales objetivos del sistema a desarrollar son la gestión de socios del gimnasio, administración de estado de cuenta de socios, administración de turnos de clase. El futuro sistema llevará el nombre de SGLP.

EL gimnasio cuenta en la actualidad con una gran cantidad de socios, que asisten a clases personalizadas, a turnos personalizados cada semana, abonando una cuota mensual.

El sistema permitirá a los encargados del gimnasio gestionar un perfil de cada cliente, donde se especifican turnos, asistencias y estado de cuenta.

El sistema debe ser escalable, permitiendo la incorporación de nuevos clientes.

El desarrollo lo llevará a cabo TucSoftware con opción a ser responsable del posterior mantenimiento del mismo.

### **III.2. Definiciones, Acrónimos y Abreviaturas**

- Definiciones:
- Acrónimos:

*GYM:* Gimnasio.

- Abreviaturas:

*IEEE:* Institute of Electrical & Electronics Engineers.

*SGLP:* Sistema de Gestión Luney Pilates.

*MLPEF:* Método de 'Luney Pilates' para la evaluación física

*DNI:* Documento Nacional de Identidad.

### III.3. Descripción General

Esta sección nos presenta una descripción general del sistema con el fin de conocer las funciones que debe soportar, los datos asociados, las restricciones impuestas y cualquier otro factor que pueda influir en la construcción del mismo.

Las funciones que debe realizar el sistema se pueden agrupar de la siguiente manera:

- **Administración de turnos:** debe permitir gestionar los turnos de las clases personalizadas, basándose en la disponibilidad de espacios del gimnasio, horarios de los profesores y cupos de los mismos.
- **Administración de socios:** debe permitir gestionar los socios de la institución desde su ingreso hasta su egreso.

Debe permitir al administrador y a cada profesor la asignación de turnos disponibles y registrar la asistencia a clases.

A su vez los usuarios participantes pueden ver estado de cuentas, turnos y clases disponibles, solo el administrador puede gestionar el estado de cuentas de cada socios.

- **Administración de cuotas:** debe permitir gestionar las cuotas de los clientes del gimnasio, pudiendo ver las cuotas pagadas y las cuotas adeudadas, de esa manera se podrá saber si el cliente puede tomar o no la/s clase/s a la/s que asiste.  
Solo el administrador del gimnasio puede actualizar la información y el estado del cliente, mientras que el profesor solo puede ver el estado, para saber si puede tomar o no la clase correspondiente. Tanto el administrador como los profesores podrán imprimir un listado de morosos a la fecha.

### III.4. Requisitos funcionales:

#### a) Gestión de Turnos

Introducción: El sistema debe permitir la creación de turnos, permitir introducir información sobre los turnos (clase, día, horario de inicio, horario de fin, sala, cupo, profesor) modificar los ya existentes y dar de baja.

Entrada: Tipo de Clase + Día + Horario de Inicio + Horario de Fin + Sala + Profesor + Cupo + Estado.

Proceso: Cuando se crea el turno se asigna un tipo de clase, un día (lunes, martes, miércoles, jueves, viernes, sábado), un horario de inicio, un horario de fin, número de sala, profesor, cupo, luego controlar disponibilidad de sala, día, horario, además de la disponibilidad del profesor revisando si hay un turno con estado activo en alguno de estos, si hay disponibilidad crear el turno y asignar estado activo. Caso contrario no registrar la operación y avisar con sus correspondientes mensajes. Si una clase se repite varias veces a la semana son turnos diferentes.

Cuando se desea dar de baja el turno, el estado pasa a estar No activo.

Salida: Datos de los turnos actualizados y mensajes de lo que está ocurriendo.

#### **b) Gestión de Socios**

Introducción: El sistema permite introducir información sobre los socios (apellido, nombres, domicilio, teléfono, DNI), modificar los ya existentes y borrarlos.

Entrada: Documento + Apellidos + Nombres + Domicilio + Teléfono + DNI

Proceso: Comprobar si se trata de un socio nuevo, dándolo de alta o actualizarlo si ya existe. No pueden existir dos socios con el mismo DNI.

Salida: Datos de socios Actualizados y mensajes de lo que está ocurriendo.

#### **c) Gestión de Profesores**

Introducción: El sistema permite introducir información sobre los Profesores (apellido, nombres, domicilio, teléfono, DNI), modificar los ya existentes y borrarlos.

Entrada: Documento + Apellidos + Nombres + Domicilio + Teléfono + DNI

Proceso: Comprobar si se trata de un profesor nuevo, dándolo de alta o actualizarlo si ya existe. No pueden existir profesores con el mismo DNI.

Salida: Datos del profesor actualizado y mensajes de lo que está ocurriendo.

#### **d) Gestión de Cuotas de Socios**

Introducción: Permite asignar las cuotas de los socios y registrar los pagos de las mismas.

Entrada: Documento + Mes + Importe + Fecha de Pago + Fecha de vencimiento

Proceso: Mensualmente se crea una cuota con el campo Fecha de pago vacío. Al Momento del pago Comprobar si la cuota ya está pagada, revisando si el campo Fecha de Pago está lleno, cancelar la operación y mostrar el respectivo mensaje de error. Comprobar si existe deuda anterior y avisar, si la cuota a pagar es la que se adeuda, registrar el pago correspondiente y avisar.

Salida: Datos de cuotas creados o actualizados y mensajes de lo que está ocurriendo.

#### **e) Gestión de asignación Turnos**

Introducción: Para asignar turnos a los Socios, los mismos deben tener libre deuda, es decir, tener la cuota mensual pagada al día, para ello éste debe estar dado de alta en el sistema de gestión de Socios.

También debe haber turnos y cupos disponibles. El sistema también permite modificar los ya existentes y borrarlos.

Entrada: Código de Alumno + Código de Turno.

Proceso: Comprobar si el socio es moroso, sacar y avisar. Caso contrario comprobar mediante la diferencia entre cupo y la cantidad de inscriptos en el turno deseado, además controlar que él no esté el mismo o en otro turno a la misma hora y día. Generar la asignación del turno.

Salida: Datos actualizados del Turno con el mensaje correspondiente.

#### **f) Listado de Morosos**

Introducción: Permite listar los Socios morosos.

Entrada: Tipo de Búsqueda (Fecha o socio) + Cadena de Búsqueda.

Proceso: Armar el listado con el socio moroso o los socios morosos y dar opción a imprimirlo.

Salida: Listado {Documento + Apellidos + Nombres + Fecha último pago}

#### **g) Listado de Socios**

Introducción: Permite listar los socios del gimnasio.

Entrada: Tipo de Búsqueda (Fecha o socio) + Cadena de Búsqueda.

Proceso: Armar un listado de los socios según el tipo de búsqueda y la cadena de búsqueda ingresada y dar la opción a imprimirlo.

Salida: Listado {Documento + apellido + nombres}

#### **h) Listado de Turnos**

Introducción: Permite listar los Turnos.

Entrada: tipo de búsqueda (clase o día) + cadena de búsqueda

Proceso: Armar un listado de los turnos y la cantidad de socios inscriptos según el tipo de búsqueda y la cadena de búsqueda ingresada y dar la opción a imprimirlos.

Salida: listado {clase, día, horario de inicio, horario de fin, sala, cupo, profesor, cantidad de inscriptos}

#### **i) Listado de socios inscriptos en un turno**

Introducción: Permite listar los socios inscriptos en un turno.

Entrada: tipo de búsqueda(turno) + cadena de búsqueda

Proceso: Armar un listado de los socios de un turno según el tipo de búsqueda y la cadena de búsqueda ingresada y dar la opción a imprimirlos.

Salida: listado {apellidos, nombres}

#### **j) Listado de profesores**

Introducción: Permite listar los profesores del gimnasio.

Entrada: tipo de búsqueda (fecha, profesor) + cadena de búsqueda

Proceso: Armar un listado de los profesores según el tipo de búsqueda y la cadena de búsqueda ingresada y dar la opción a imprimirlos.

Salida: listado {apellido, nombre, documento}

### **III.5. Suposiciones y Dependencias**

Suposiciones:

Se asume que los requisitos en este documento son estables una vez que sean aprobados por la Dirección de Luney Pilates. Cualquier petición de cambios en la especificación debe ser aprobada por Todas las partes intervinientes y será gestionada por el equipo de desarrollo.

Dependencias: No la tiene.

### **III.6. Requisitos de Usuario y Tecnológicos**

Requisitos de usuario:

Los usuarios serán el administrador y los profesores del gimnasio 'Luney Pilates'. Las interfaces deben ser intuitivas, fáciles de usar y amigables.

Requisitos tecnológicos:

En vista de que es necesario instalar las aplicaciones en varias computadoras, y teniendo en cuenta un futuro crecimiento del gimnasio, se ha optado por un entorno económico y fácil de instalar. La aplicación se ejecutará sobre un esquema cliente/servidor, con los procesos e interfaz de usuario ejecutándose en los clientes y éstos solicitando requerimientos al servidor que cumple su proceso.

El sistema operativo de los clientes será Microsoft Windows 7 o superior. El sistema operativo del servidor será Microsoft Windows server 2012 R2. Requisitos fijados por la configuración actual de red.

La Red actual tiene una topología estrella y tecnología Ethernet.

### **III.7. Requisitos de Interfaces Externas**

#### Interfaces de usuario:

La interfaz de usuario debe ser orientada a ventanas tipo Windows.

#### Interfaces Hardware:

Ratón y teclado estándar, Monitor, impresora.

#### Interfaces software:

### **III.8. Requisitos de Rendimiento**

El tiempo de respuesta de la aplicación a cada función solicitada por el usuario no debe ser superior a los 3 segundos. El tiempo de respuesta a los listados dependerá de la tecnología de impresión.

### **III.9. Requisitos de Desarrollo**

El ciclo de vida será el de Prototipado Evolutivo, debiendo orientarse hacia el desarrollo de un sistema flexible que permita incorporar de manera sencilla cambios y nuevas funcionalidades.

### **III.10. Restricciones de Diseño**

Ajuste a estándares: Interfaz de usuario Basada en Bootstrap.

Seguridad: La seguridad de los datos será establecida por el Sistema Gestor de Base de Datos Relacional que se emplee.

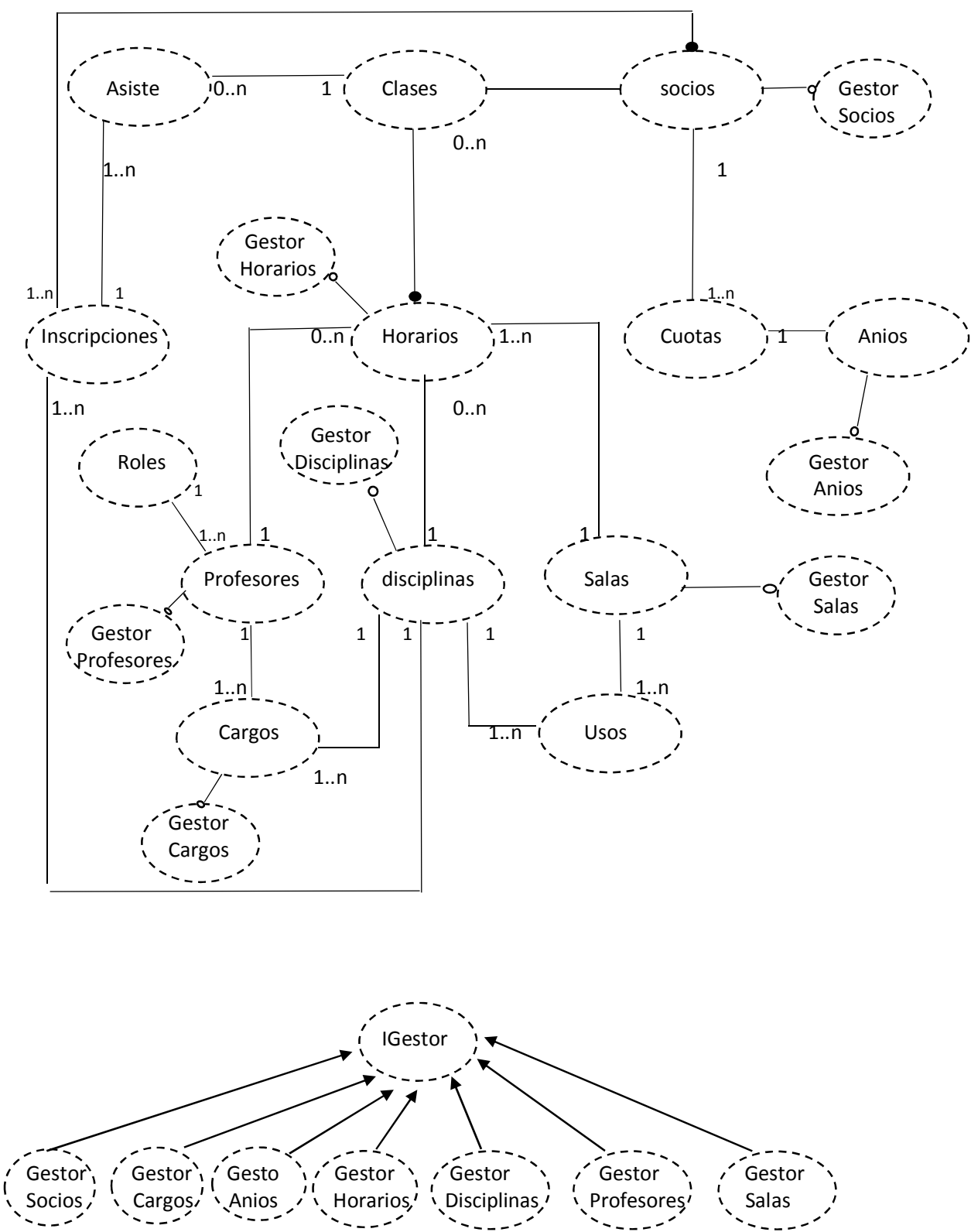
Política de Respaldo: El administrador llevará a cabo un respaldo de datos en discos externos una vez cada 3 meses. El motor de Base de datos configurado para realizar backups una vez al día. Conservar los respectivos archivos de respaldo de los últimos 7 días y también guardar un backup de inicio de semana.

#### Base de Datos:

El Sistema Gestor de Base de Datos debe ser relacional y se accederán a los mismos usando la tecnología innoDB.

Política de Borrado: No se ha definido.

Diagrama de clases





Fichas Tecnicas de Cases:

Clase Socios	
Atributos	Operaciones
- IdSocio: Entero + Nombre: Cadena +Apellido: Cadena +Documento: Cadena +Teléfono: Cadena +Fecha de Nacimiento: Fecha +Dirección: Cadena +Estado: Cadena	Cadena Pago() Cadena generarCuotas() ConjuntoCuotas ListarCuotas(estado:Cadena,fechaDesde:Fecha,FechaHasta:Fecha) ConjuntoInscripciones ListarInscripciones() Cadena InscribirDisciplina(d:Disciplinas) Boolean esMoroso()

Clase Clases	
Atributos	Operaciones
-IdClases: Cadena -IdHorarios: Cadena +Fecha: Fecha	

Clase Asiste	
Atributos	Operaciones
-IdSocios: Entero -IdClases: Entero +Fecha Asistencia: Fecha +Asistencia:boolean	

Clase Inscripciones
---------------------

Atributos	Operaciones
-IdSocio: Entero -IdDisciplina: Entero +Estado: Cadena	ConjuntoAsistencias ListarAsistencias() ConjuntoDisciplinas ListarDisciplinasInscriptas()

Clase Horarios	
Atributos	Operaciones
- IdHorarios: Entero + Dia de la Semana: Cadena + Hora Inicio: Cadena + Hora Fin: Cadena	ConjuntoClases ListarClases() ConjuntoProfesores ListarProfesores() ConjuntoSalas ListarSalas() ConjuntoDisciplinas ListarDisciplinas()

Clase Anios	
Atributos	Operaciones
- IdAnios: Entero + Anio: Cadena + Mes: Cadena	Cadena GenerarCuotasAnuales()

Clase Cuotas	
Atributos	Operaciones
- IdSocio: Entero - IdAnio:Entero - IdCuota:Entero + Estado: Cadena + Monto: Decimal + Fecha Vencimiento: Fecha + Fecha Pago: Fecha	

Clase Disciplinas	
Atributos	Operaciones
- IdDicipilina: Entero + Nombre: Cadena + Estado: Cadena	ConjuntoSalas ListarSalas() ConjuntoProfesores ListarProfesores() ConjuntoHorarios ListarHorarios()

Clase Salas	
Atributos	Operaciones
- IdSalas: Entero + Nombre: Cadena + Estado: Cadena + Cupo: Entero	ConjuntoHorarios VerHorarios() ConjuntoDisciplinas VerDisciplinaDeLaSala()

Clase Usos	
Atributos	Operaciones
<ul style="list-style-type: none"> <li>+ IdSalas: Entero</li> <li>+ IdDicipilina: Entero</li> </ul>	

Clase Profesores	
Atributos	Operaciones
<ul style="list-style-type: none"> <li>- IdProfesor: Entero</li> <li>+ Nombre: Cadena</li> <li>+ Apellido: Cadena</li> <li>+ Documento: Entero</li> <li>+ Domicilio: Cadena</li> <li>+ Telefono: Entero</li> <li>+ Estado: Cadena</li> <li>+ Contraseña: Cadena</li> </ul>	Cadena ListarHorarios() ConjuntoCargos CargosActicos() Cadena AsignarRoles(r:Roles)

Clase Roles	
Atributos	Operaciones
<ul style="list-style-type: none"> <li>- IdRoles: Entero</li> <li>+ Rol: Cadena</li> </ul>	

Clase Cargos	
Atributos	Operaciones
<ul style="list-style-type: none"> <li>- IdCargo:Entero</li> <li>+ IdProfesor: Entero</li> <li>+ IdDicipilina: Entero</li> </ul>	ConjuntoProfesores ListarProfesoresPorCargos()

Las clases Gestor Rubros, Gestor Proveedores, Gestor Reparticiones, Gestor Usuarios poseen únicamente las operaciones heredadas de la metaclasselGestor.

IGestor	
Atributos	Operaciones
	Cadena alta(objeto o) Cadena borrar(objeto o) Cadena modificar(objeto o) Cadena buscar(objeto o) ConjuntoObjeto Listar()

Clase: Gestor Socios	
Atributos	Operaciones
	ConjuntoSocios ListarMorosos()

Clase Gestor Cargos	
Atributos	Operaciones

Clase GestorHorarios	
Atributos	Operaciones
	ConjuntoHorarios ListarTurnos()

Clase Gestor Anios	
Atributos	Operaciones

Clase Gestor Disciplinas	
Atributos	Operaciones

Clase Gestor Salas	
Atributos	Operaciones

Clase Gestor Profesores	
Atributos	Operaciones

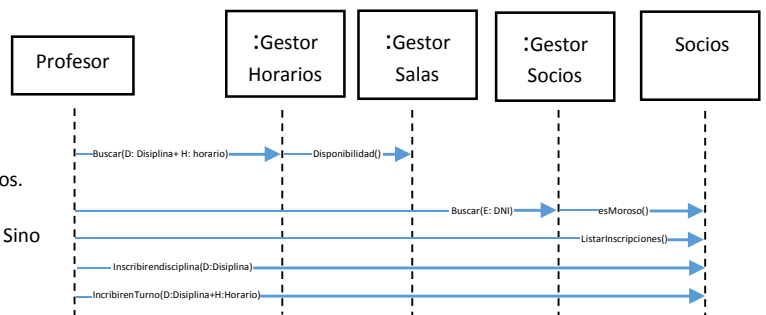
## V. Casos de Usos

### V.1 Lista de casos de usos:

CU01	Crear Horario
CU02	Reservar Turno
CU03	Modificar Horario
CU04	Dar de Baja el Horario
CU05	Listar Turnos
CU06	Nuevo Socio
CU07	Modificar Socio
CU08	Dar de Baja Socio
CU09	Buscar Socio
CU10	Listar Socios Inscriptos en Cada Turno
CU11	Nuevo Profesor
CU12	Modificar Profesor
CU13	Dar de Baja Profesor
CU14	Buscar Profesor
CU15	Listar Profesores
CU16	Registrar Pago de Cuota
CU17	Asignar Cuotas al Socio
CU18	Cancelar Deuda
CU19	Imprimir Listado de Morosos
CU20	Imprimir Listado de Socios
CU21	Imprimir Lista Turnos

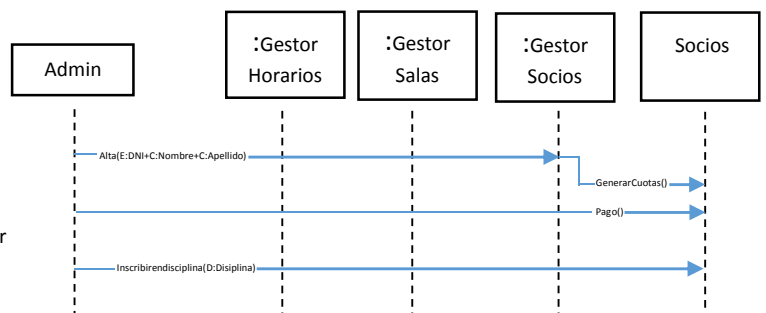
#### CU02. Reservar Turno

1. El socio indica el horario y la clase que desea
2. Buscar Turnos Disponibles
3. Pedir DNI al Socio en caso de no tener sus datos.
4. Corroborar que no sea moroso.
5. Corroborar que esté inscripto en la Disciplina, Sino  
5.1 Inscribir en la disciplina
6. Reservar Turno



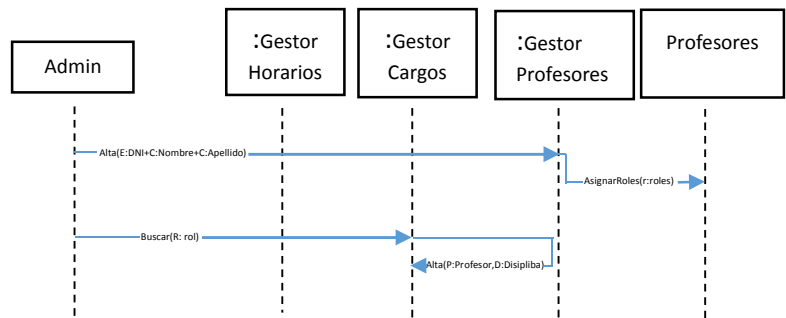
#### CU06. Nuevo Socio

1. Pedir datos al nuevo socio
2. Dar de Alta al nuevo Socio
3. Generar Cuotas
4. Realizar el Pago de la cuota
5. Preguntar al socio que Disciplina se va inscribir
6. Inscribir en la Disciplina
7. CU02. Reservar Turno



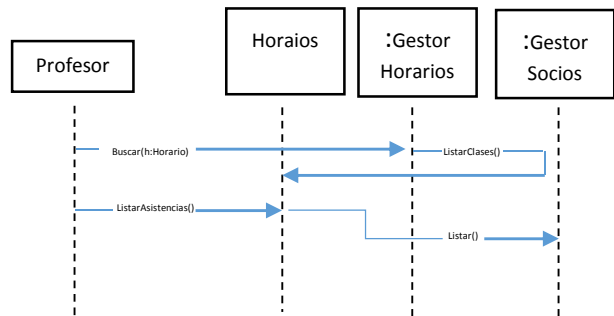
CU11. Nuevo Profesor

- 1. Pedir datos al nuevo profesor
- 2. Dar de Alta al nuevo profesor
- 3. Asignar Rol al profesor
- 4. Asignar cargo
  - 4.1 Listar Cargos Posibles según Rol
  - 4.2 Dar de alta



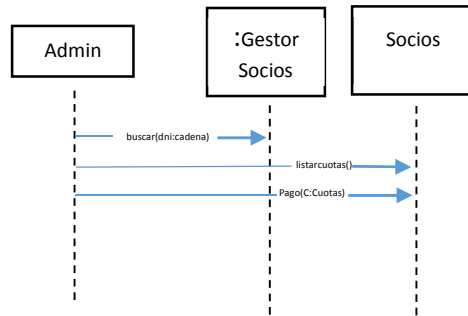
CU10. Listar Socios Inscriptos en Cada Turno

- 1. Solicitar horario del turno
- 2. Buscar horario
- 3. Buscar clase en el horario
- 4. Listar Asistencias
- 5. Listar Socios



CU16. Registrar Pago de Cuota

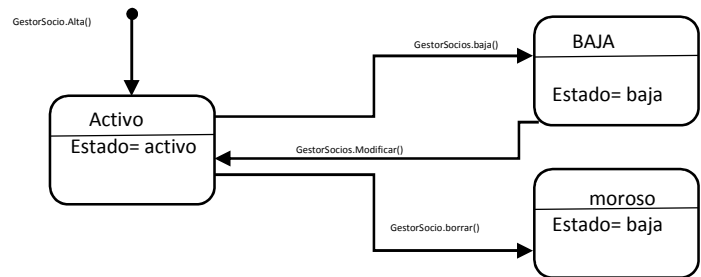
- 1. Pedir DNI al Socio
- 2. Buscar Socio
- 3. Listar cuotas adeuda
- 4. Pagar una cuota



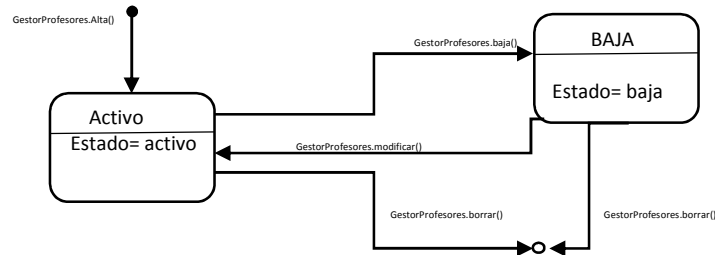


Diagramas de transición de estados.

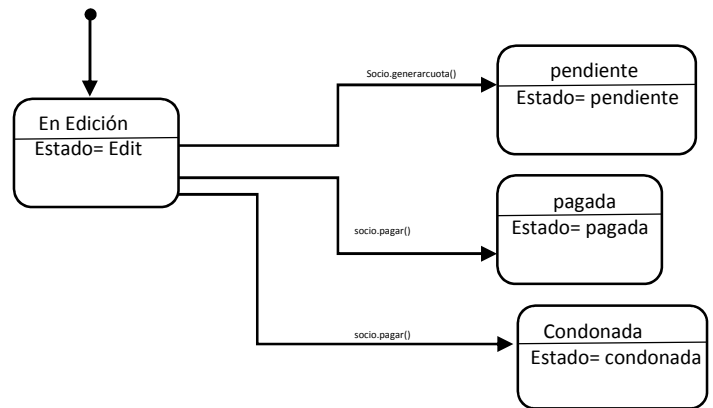
Dte Socios.



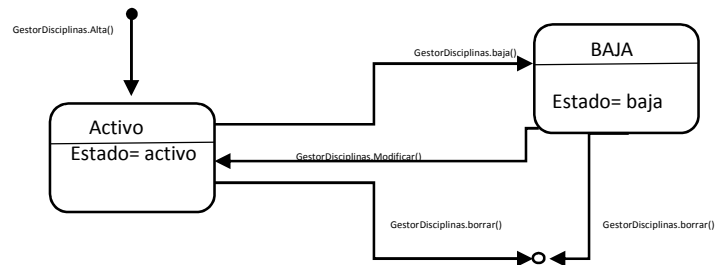
Dte Profesores.



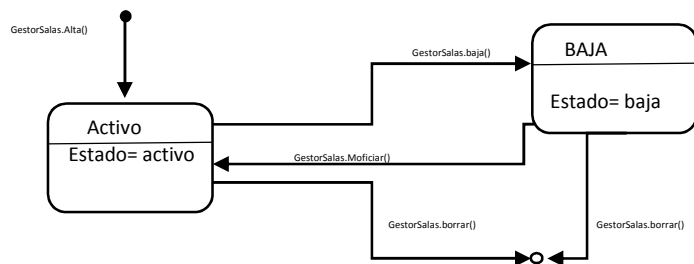
Dte Cuotas.

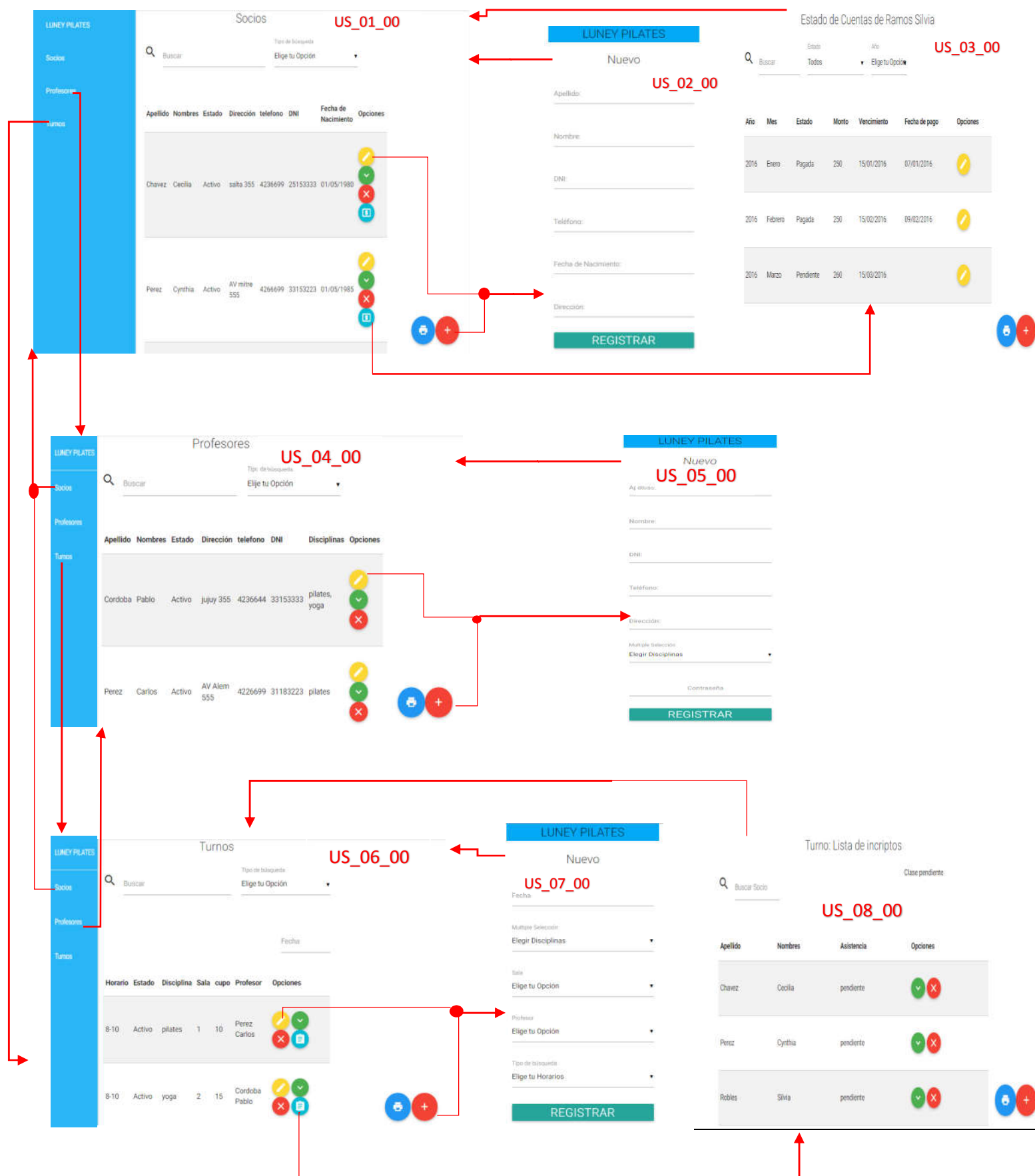


Dte Disciplinas.



Dte Salas.





Nota: Los datos de personas introducidos son ficticios, sólo de carácter ilustrativo.

Tablas de Transición de Escenarios.

La tabla de transición de escenarios del guión usuarios se detalla a continuación:

	US_01_00	US_02_00	US_03_00	US_04_00	US_05_00	US_06_00	US_07_00	US_08_00
US_01_00		US_01_6/7	US_01_08	US_01_02		US_01_03		
US_02_00	US_01_07							
US_03_00	US_01_07							
US_04_00	US_04_01				US_04_6/7	US_04_03		
US_05_00				US_05_08				
US_06_00	US_06_01			US_04_02			US_06_8/9	US_06_10
US_07_00						US_07_06		
US_08_00						US_08_04		

Escenario US\_01\_00: Socios

LUNEY PILATES

Socios01

Profesores02

Turnos03

US\_01\_00

Socios

04

05

06

09

10

08

11

07

Al cargarse la ventana se muestra en la tabla US\_01\_12 todos los datos de los socios en orden alfabético. Los botones US\_01\_02 y US\_01\_03 llevan a los escenarios US\_01\_04 y US\_01\_06 respectivamente. El campo US\_01\_04 y la lista

US\_01\_05 me permiten realizar una búsqueda de los Socios. Botones US\_01\_06 y US\_01\_07 llevan al escenario US\_02\_00 donde me permiten dar de alta un nuevo turno o modificar. Botón US\_01\_09 permite cambiar el Estado de un Socio, US\_01\_10 borrar un Socio, mientras que el botón US\_01\_08 lleva al escenario US\_03\_00 donde puedo ver el estado de cuentas de un socio. US\_01\_11 me permite imprimir la lista que se muestra en la tabla.

Escenario US\_02\_00: Socios

**US\_02\_00**

LUNEY PILATES

Nuevo

Apellido:

01

Nombre:

02

DNI:

03

Teléfono:

04

Fecha de Nacimiento:

05

Dirección:

06

REGISTRAR

07

Esta ventana se carga en dos modalidades: alta y modificación. En la primera opción, se pone el título de la ventana como nuevo, se ponen vacíos los cuadros de texto US\_02\_01/2/3/4/5/6. En la segunda opción, se llenan los campos US\_02\_01/2/3/4/5/6 con los datos del socio que se quiere modificar. La lista desplegable US\_04\_02 tiene los tipos de teléfonos más usados (particular, casa, trabajo, celular y fax). Al presionar el botón US\_02\_07 (Registrar), se vuelve a la ventana US\_01\_00 con los datos del registro para ser guardado o modificado, mientras que si se presiona el botón retroceder del navegador en uso, se vuelve al escenario US\_01\_00 sin hacer nada.

US\_03\_00

Estado de Cuentas de Ramos Silvia

01

02

03

Q

Buscar

Estado

Todos

Año

Elige tu Opción

07

Año	Mes	Estado	Monto	Vencimiento	Fecha de pago	Opciones
2016	Enero	Pagada	250	15/01/2016	07/01/2016	<div>04</div>
2016	Febrero	Pagada	250	15/02/2016	09/02/2016	
2016	Marzo	Pendiente	260	15/03/2016		

06

05

Al cargarse la ventana se muestra en la tabla US\_03\_07 todas las cuotas en orden según el mes. El boton US\_03\_05 me permite generar las cuotas del siguiente año o las del año en transcurso en caso de que sea un socio nuevo. El campo US\_03\_01 me permiten realizar una búsqueda de las cuotas según fecha de pago. Botones US\_03\_02 y US\_03\_03 me permite filtrar los resultados de la tabla según el estado de las cuotas y el año respectivamente llevan al escenario. US\_03\_06 me permite imprimir la lista que se muestra en la tabla.

LUNEY PILATES

Socios01

Profesores02

Turnos03

US\_04\_00

Profesores

05

04

05

04

05

06

07

08

09

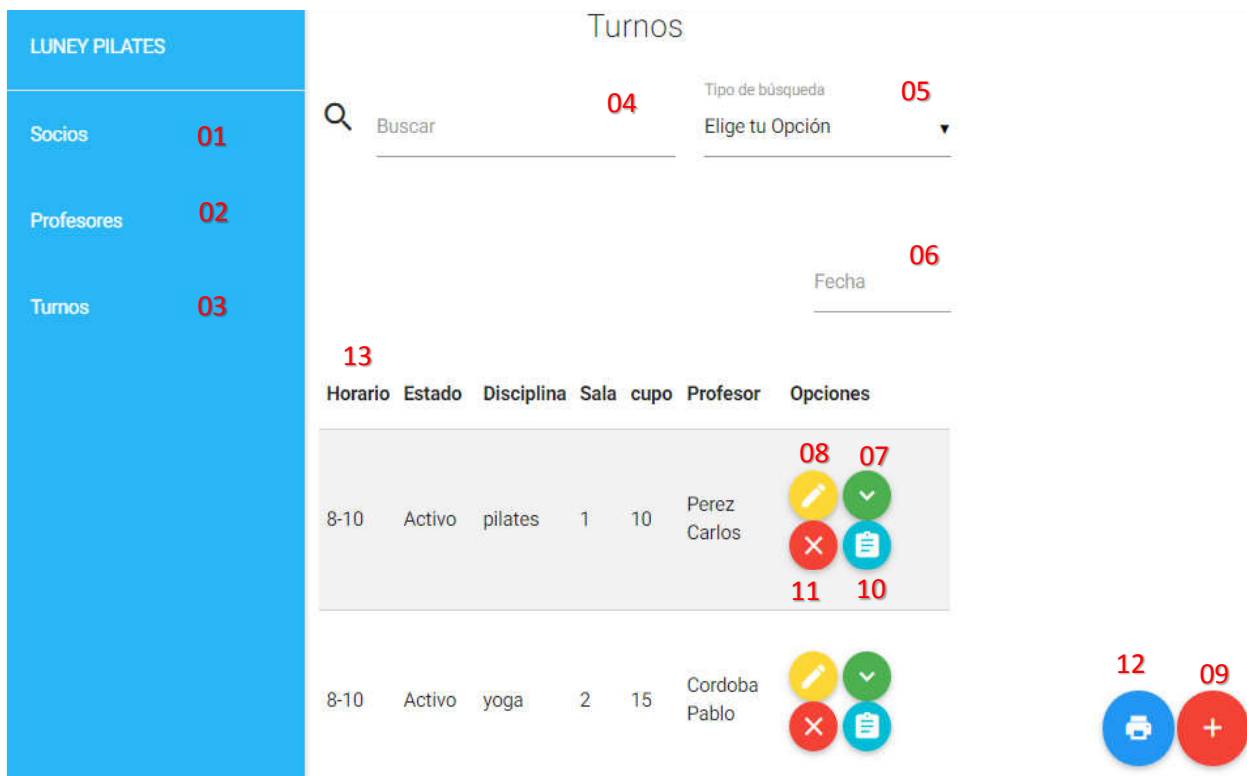
07

10

Apellido	Nombres	Estado	Dirección	telefono	DNI	Disciplinas	Opciones
Cordoba	Pablo	Activo	jujuy 355	4236644	33153333	pilates, yoga	<div>06</div> <div>07</div> <div>08</div>
Perez	Carlos	Activo	AV Alem 555	4226699	31183223	pilates	<div>09</div> <div>07</div>

Al cargarse la ventana se muestra en la tabla US\_04\_10 todos los profesores en orden alfabético. Los botones US\_04\_02 y US\_04\_03 llevan a los escenarios US\_04\_00 y US\_06\_00 respectivamente. El campo US\_04\_04 y la lista US\_04\_05 me permiten realizar una búsqueda de los profesores. Botones US\_04\_06 y US\_04\_07 llevan al escenario US\_05\_00 donde me permiten dar de alta un nuevo profesor o modificar. Botón US\_04\_07 permite cambiar el Estado de un profesor, US\_04\_08 borrar un profesor. US\_04\_10 me permite imprimir la lista que se muestra en la tabla.

## US\_06\_00



**LUNEY PILATES**

**Turnos**

01 Socios

02 Profesores

03 Turnos

04 Buscar

05 Tipo de búsqueda  
Elige tu Opción

06 Fecha

13

Horario	Estado	Disciplina	Sala	cupo	Profesor	Opciones
8-10	Activo	pilates	1	10	Perez Carlos	08 07 11 10
8-10	Activo	yoga	2	15	Cordoba Pablo	08 07 11 10

12 09

Al cargarse la ventana se muestra en la tabla US\_06\_13 todos los turnos en orden según el horario. Los botones US\_06\_01 y US\_06\_03 llevan a los escenarios US\_01\_00 y US\_04\_00 respectivamente. El campo US\_06\_04 y la lista US\_06\_05 me permiten realizar una búsqueda de los profesores. Botones US\_06\_08 y US\_06\_09 llevan al escenario US\_07\_00 donde me permiten dar de alta un nuevo turno o modificar. Botón US\_06\_07 permite cambiar el Estado de un turno, US\_06\_11 borrar un turno, mientras que el botón US\_06\_10 lleva al escenario US\_08\_00 donde puedo ver la lista de socios inscriptos en un turno. US\_06\_12 me permite imprimir la lista que se muestra en la tabla.

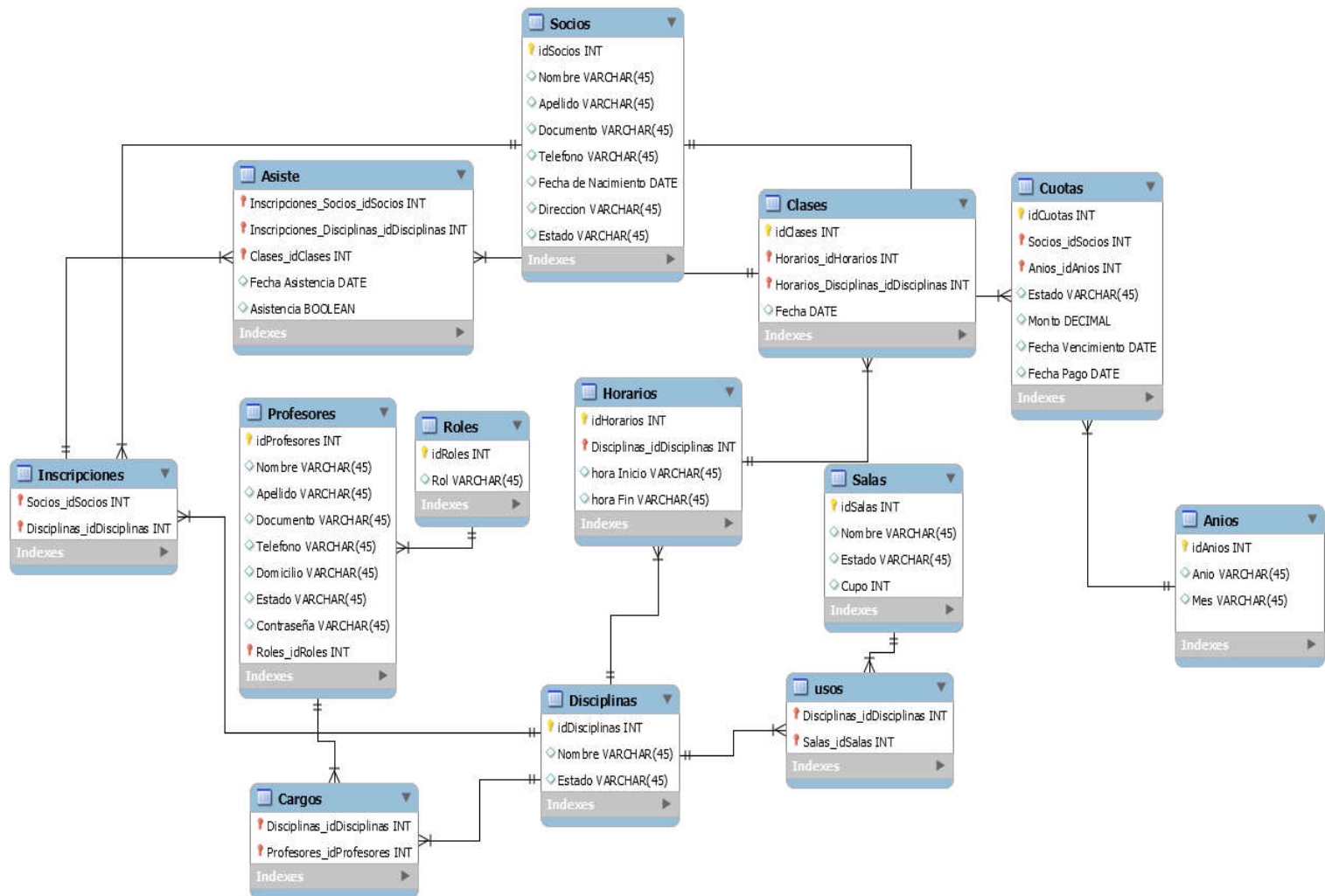
### Fichas de Objetos de Escenario.

La nomenclatura de identificación de escenarios y objetos se describe a continuación: dos letras que identifican al usuario al que pertenece el escenario; luego siguen dos números que identifican al escenario; finalmente dos números que identifican al objeto del escenario. Así el objeto US\_02\_21, será el objeto 21 del escenario 02 del guión usuario. El objeto 00 se reserva para el escenario propiamente dicho, que también es susceptible de interactuar con el usuario en forma de eventos. Para la descripción de los objetos de entorno se adoptan estándares de tipos de objetos con similares características para facilitar la descripción. Estos estándares contienen las propiedades y los valores comunes a todos los objetos. Se utilizó el framework materializecss (<http://materializecss.com/>) con hojas de estilos css predefinidas para realizar la interfaz.





## Diagrama Realacional:



## Arquitectura Fisica del sistema:

La Arquitectura del sistema será de Modelo Vista Controlador (MVC). Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Esta arquitectura presenta las siguientes ventajas:

- Facilita la administración de la complejidad, al dividir una aplicación en el modelo, la vista y el controlador.
- Bajo costo de mantenimiento.
- Seguridad

### Componentes de MVC se podrían definir como sigue:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.<sup>12</sup>
- El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- La Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

### Interacción de los componentes:

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por

ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro

5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

## Ficha técnicas de eventos de objetos de escenario

### US\_01\_00: Socios

Nombre	US_01_00	Evento	Cargar Pagina
Llenar lista US_01_12 con Socios. Buscar() Si Existen elementos en la lista, entonces: Completar la lista con los datos de los socios y crear los botones US_01_06-08-09-10 para cada socio. Si NO existen elementos en la tabla, entonces: Completar US_01_12 con el mensaje "No existen socios en la base datos", no generar los botones US_01_06-08-09-10, e inhabilitar el botón US_01_05.			

Nombre	US_01_05	Evento	Click
Una vez seleccionada una opción (Nombre,DNI) habilita el campo de búsqueda US_01_04, Pudiéndose completar solamente con letras en el caso de ser Nombre, o solo con números en el caso de ser DNI. El campo de búsqueda directamente no escribe nada, en el caso de ingresar un dato no perteneciente al grupo.			

Nombre	US_01_04	Evento	Presionar Tecla
Si la tecla Presionada es ENTER: Llenar la lista US_01_12 con Socios.Buscar(US_01_04) Si Existen elementos en la lista, entonces: Completar la lista con el/los dato/s de/l los socios y crear los botones US_01_06-08-09-10 para el/cada socio que coincida con la cadena de búsqueda. Si NO existen elementos en lista, entonces: Completar US_01_12 con el mensaje "No existen socios en la base datos", no generar los botones US_01_06-08-09-10			

## US\_02\_00: Socios

Nombre	US_02_00	Evento	Cargar Pagina
<p>Si US_02_00 posee argumento s: Socio &lt;&gt; ""</p> <p>Se considera una modificación, entonces el campo:</p> <p>US_02_01 = Socio.apellido</p> <p>US_02_02 = Socio.nombre</p> <p>US_02_03 = Socio.dni</p> <p>US_02_04 = Socio.telefono</p> <p>US_02_05 = Socio.fachaNacimiento</p> <p>US_02_06 = Socio.direccion</p> <p>Si US_02_00 NO posee argumento, entonces los campos: US_02_01-02-03-04-05-06 se dejan en blanco y se considera, altaSocio.</p>			

Nombre	US_02_07	Evento	Click
<p>Si los campos US_02_01 al US_02_06 &lt;&gt; "", entonces:</p> <p>Verificar que cada campo este completo correctamente</p> <p>Crear nuevo Socio en la base de datos.</p> <p>Si los campos US_02_01 al US_02_06 == "" o no están completos correctamente, entonces:</p> <p>Mostrar mensaje de Error.</p>			

## Fichas técnicas de métodos de objetos.

### Clase Horarios

Método:	ConjuntoClases ListarClases()
<p>Conectarse con base de datos</p> <p>Para cada clase de la tabla clases ordenada alfabéticamente nombre hacer:</p> <p>Agregar Clase a ListarClases</p>	

### Clase: Gestor Socios

Método:	Cadena alta(s: Socio)
<p>Conectarse con base de datos</p> <p>alta ="OK"</p> <p>Si ocurre en error un error en la comunicación o en los datos, entonces:</p> <p>alta ="Mensaje de Error"</p> <p>Sino</p> <p>Agregar registro a la tabla socios</p>	

Método:	ConjuntoSocios ListarMorosos()
Conectarse con base de datos Para cada Socio de la tabla Socios: Si socio.esMoroso()=="True" Agregar socio a ListarMorosos.	