



Diseño de Sistemas

Trabajo Practico Integrador 2

2018

Profesores:

- Gaona, Germán
- Sandoval Verón, Valeria

Grupo 5 - Tesla

- Alegre, Nicolás
- Enrique, Gerardo
- López, Sebastián
- Solís, Santiago
- Zimmermann, Federico

Fecha de Entrega: 05/12/2018

ÍNDICE

Diagrama de casos de uso general	2
Diagrama de casos de uso simplificado	3
Realización de caso de uso: Añadir Clase	4
Descripción de caso de uso	4
Tarjetas CRC	5
Diagrama de Clases	9
Diagrama de Secuencia	10
Realización de caso de uso: Obtener informe de asistencia	11
Descripción de caso de uso	11
Tarjetas CRC	12
Diagrama de Clases	16
Diagrama de Secuencia	17
Realización de caso de uso: Registrar Socio	18
Descripción de caso de uso	18
Tarjetas CRC	19
Diagrama de clases	23
Diagrama de Secuencia	24
Implementación	25
Capturas de pantalla	29

DIAGRAMA DE CASOS DE USO GENERAL

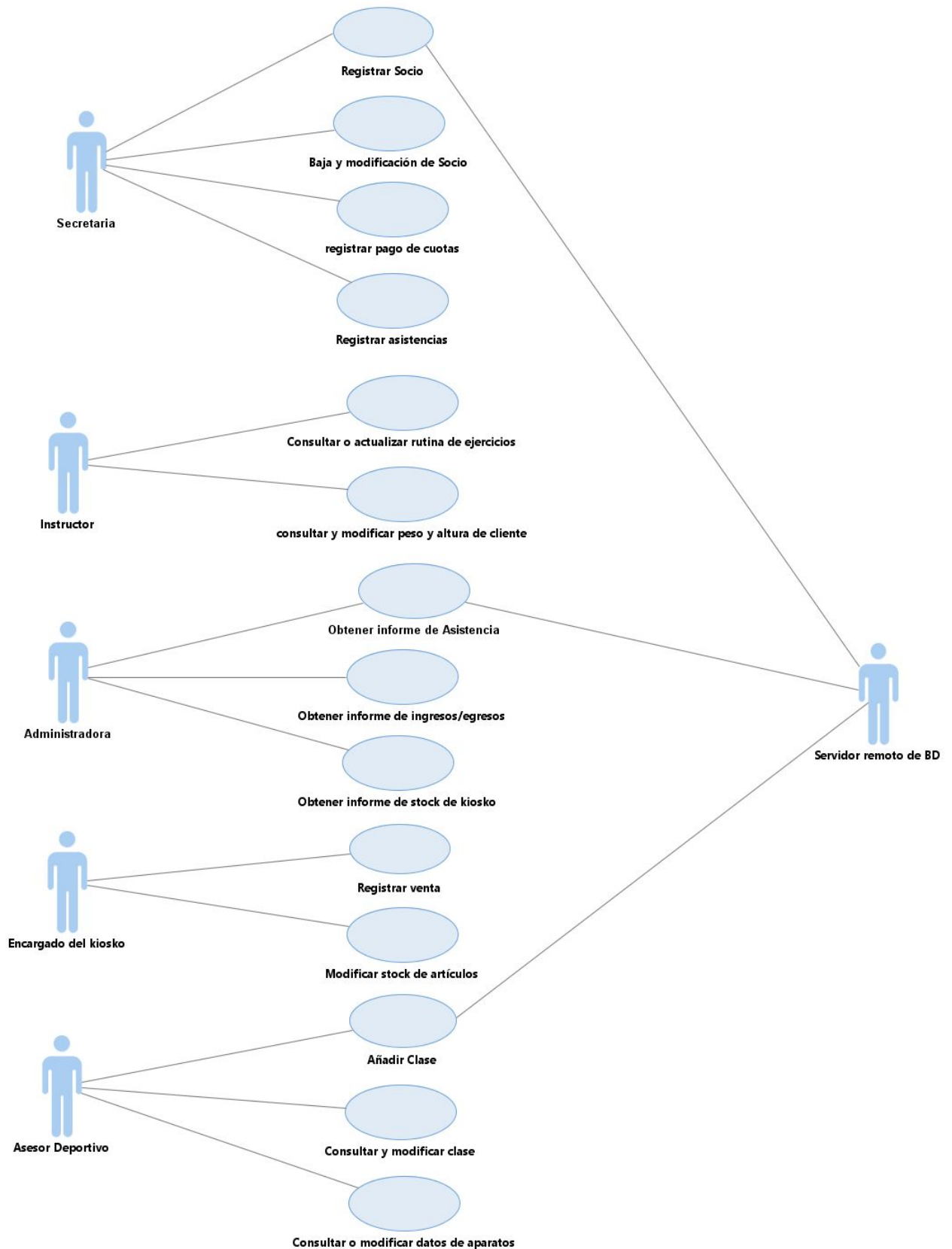
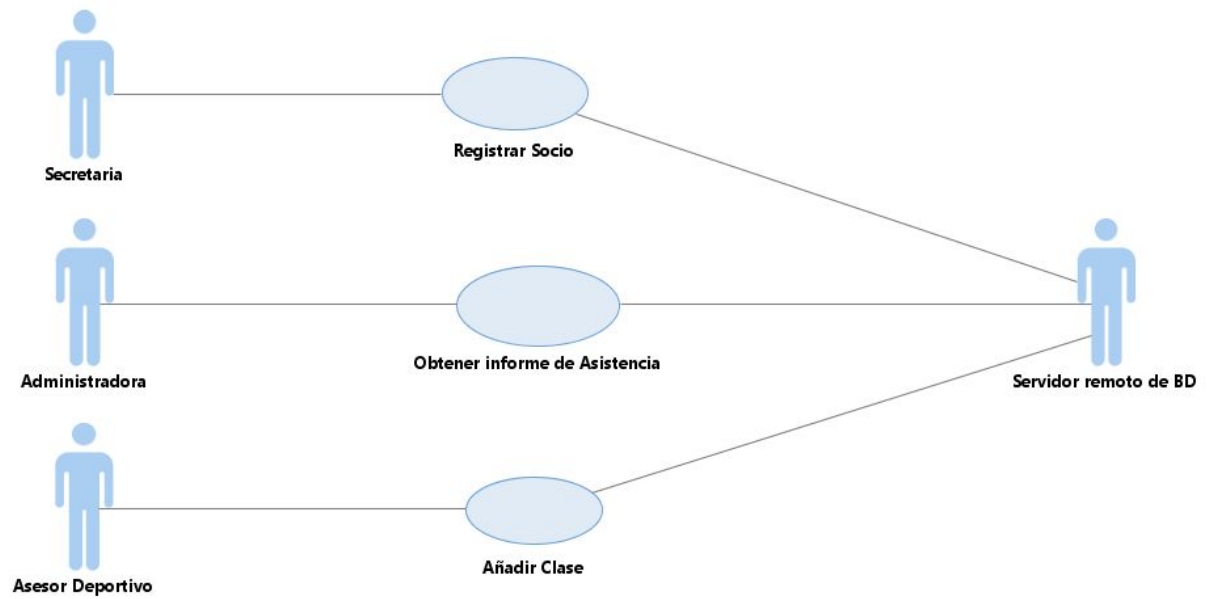
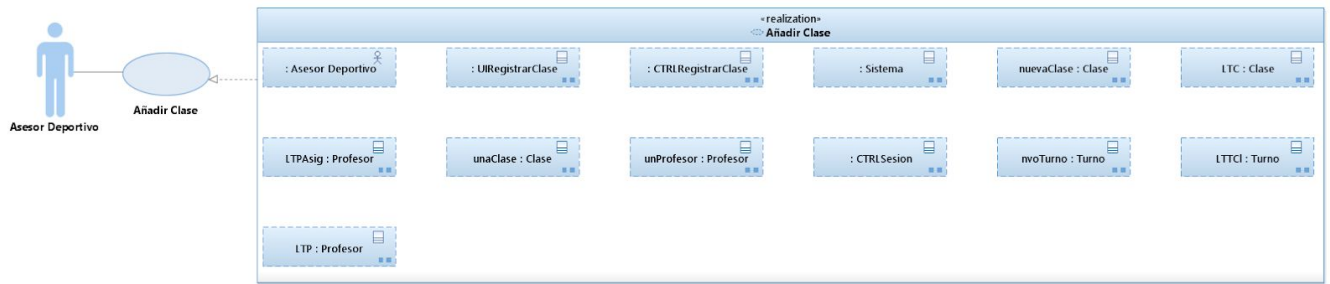


DIAGRAMA DE CASOS DE USO SIMPLIFICADO



REALIZACION DE CASO DE USO: Añadir clase



• DESCRIPCIÓN DE CASO DE USO

Descripción de caso de uso: Añadir clase

Actor: Asesor deportivo.

Actor secundario: Servidor remoto de base de datos.

Precondición: El asesor deportivo debe estar autenticado y el servidor remoto de base de datos debe estar disponible.

Postcondición: Clase añadida.

Camino Estándar:

1. El usuario selecciona la opción "Registrar clase"
2. El sistema solicita el nombre de la clase.
3. El usuario ingresa el nombre de la clase.
4. El sistema solicita el monto mensual de la clase.
5. El usuario ingresa el monto mensual.
6. El sistema solicita días y horarios del turno a agregar.
7. El usuario ingresa días y horarios.
8. El sistema muestra los profesores de la sucursal correspondiente a la del asesor autenticado con disponibilidad en el turno ingresado.
9. El usuario selecciona el profesor que desea asignar al turno.
10. El sistema asigna el profesor seleccionado a dicho turno, y a este último lo guarda en la nueva clase.
11. El usuario confirma la creación de la clase.
12. El sistema guarda la clase.

Fin de Caso de Uso

Camino Alternativo:

8.a Profesores no disponibles

8.a.1. El sistema informa que no existen profesores disponibles de esa sucursal en ese horario.

Fin de caso de uso

9.a. El usuario desea ingresar mas profesores al turno.

9.a.1. El usuario selecciona todos los profesores que desea asignar al turno.

9.a.2. El sistema asigna los profesores seleccionados a dicho turno, y a este último lo guarda en la nueva clase.

9.a.3. Retorna al paso 11 del camino estándar.

11.a. El usuario desea agregar más turnos.

11.a.1. Retorna al paso 6 del camino estándar.

- TARJETAS CLASE-RESPONSABILIDAD-COLABORACIÓN (CRC)

SISTEMA	
Propósito: Representa a la clase principal que gestiona el sistema.	
Responsabilidad	Colaboración
<p>Conocer información sobre los profesores del gimnasio.</p> <p>Conocer información de las clases ofrecidas.</p>	<p>Profesor</p> <p>Clase</p>

PROFESOR	
Propósito: Representa a una persona que instruye al cliente en sus actividades físicas.	
Responsabilidad	Colaboración
<p>Almacenar información acerca de los profesores que dictan las clases.</p> <p>Conoce los turnos de la clase a los que está asignado.</p>	Turno

CLASE	
Propósito: Representa una actividad llevada a cabo en el gimnasio.	
Responsabilidad	Colaboración
<p>Almacenar información de una clase</p> <p>Conocer los turnos en los cuales se dicta.</p>	Turno

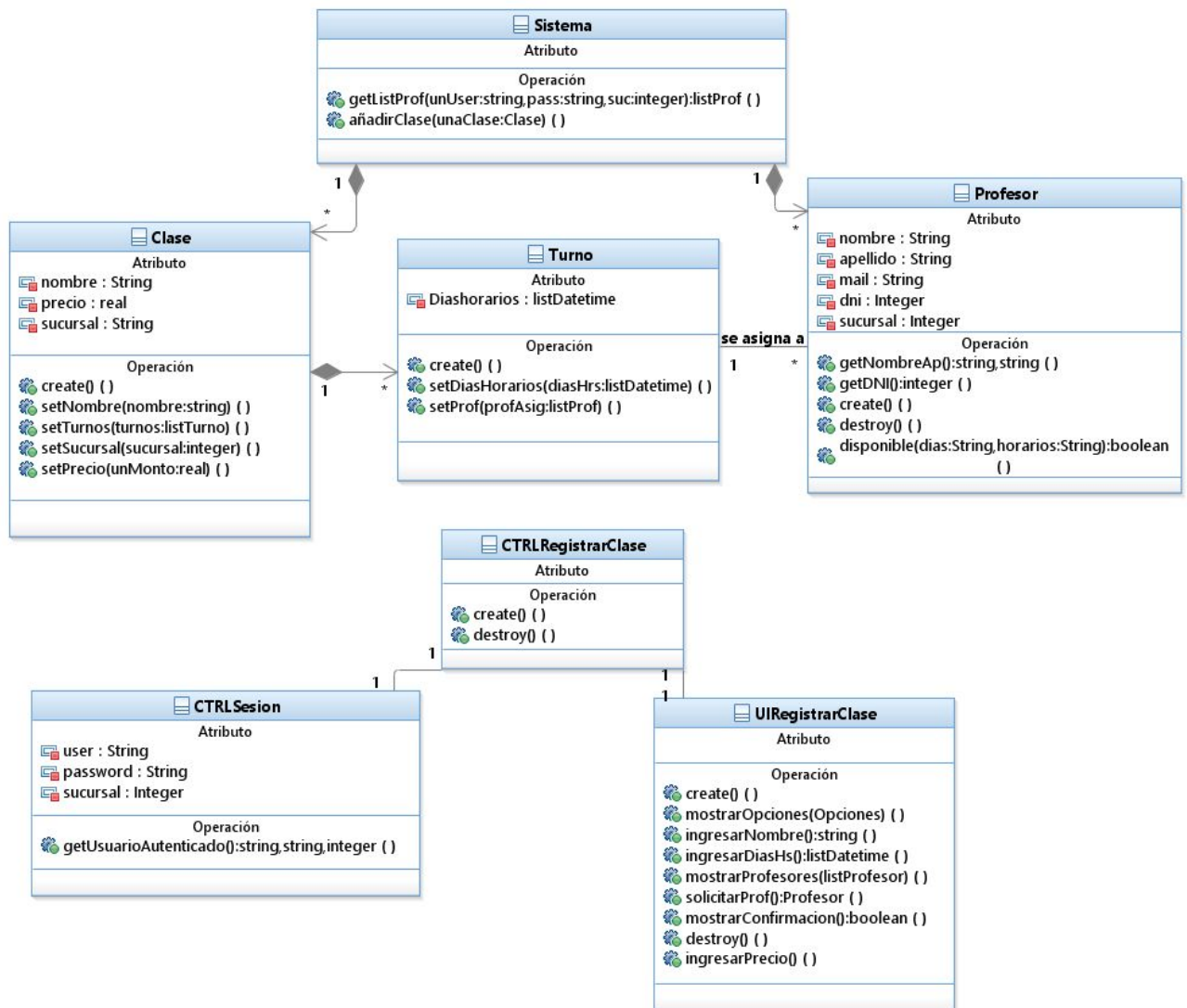
TURNO	
Propósito: Representa un turno en el que se dicta una clase	
Responsabilidad	Colaboración
<p>Almacenar información de los turnos en los que se dictan las clases.</p> <p>Conocer los profesores que están asignados a cada turno.</p>	Profesor

UIRegistrarClase	
Propósito: Representa la interfaz entre el actor y el Sistema para registrar una clase	
Responsabilidad	Colaboración
<p>Recibir los datos de la clase que se quiere registrar.</p> <p>Recibir y enviar mensajes de confirmación.</p> <p>Mostrar en el formato adecuado las opciones a completar para registrar una clase.</p>	CTRLRegistrarClase

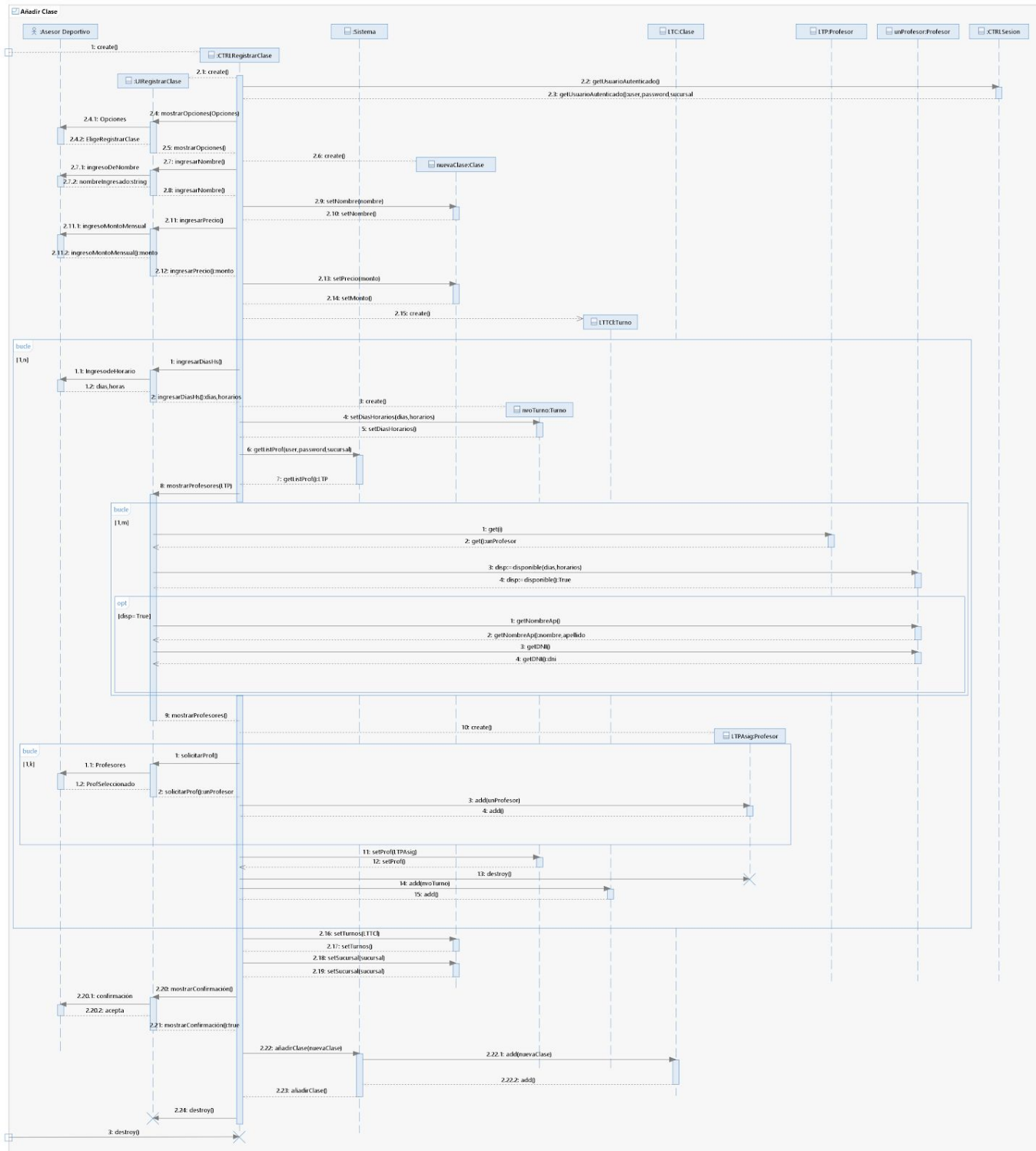
CTRLRegistrarClase	
Propósito: Representa el controlador del CU que envía las acciones a ser realizadas al sistema	
Responsabilidad	Colaboración
<p>Crear la UI</p> <p>Enviar mensajes al sistema sobre las acciones que fueron seleccionadas por el actor</p> <p>Recibir los resultados de las operaciones realizadas por el sistema</p>	<p>UIRegistrarClase</p> <p>Sistema</p> <p>CTRLsesion</p>

CTRLSesion	
Propósito: Representa el controlador de sesiones	
Responsabilidad	Colaboración
<p>Permite conocer el usuario autenticado en cada momento y la sucursal a cual corresponde.</p>	<p>CTRLRegistrarClase</p>

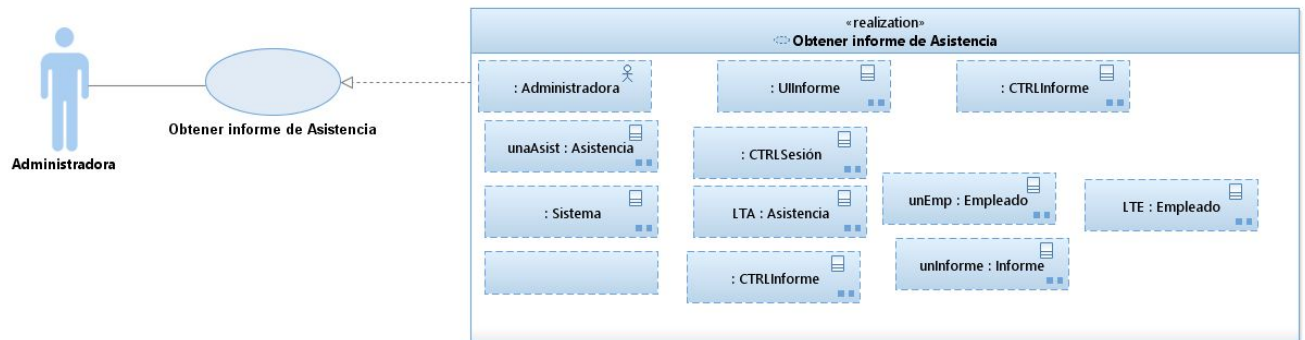
- DIAGRAMA DE CLASES



• DIAGRAMA DE SECUENCIA



REALIZACION DE CASO DE USO: Obtener informe de asistencia



• DESCRIPCIÓN DE CASO DE USO

Descripción del CU: Obtener informe de asistencia

Actor primario: Jefa de Administración

Actor secundario: Servidor remoto de base de datos.

Precondición: la jefa de administración debe estar autenticada y el servidor remoto de base de datos debe estar disponible.

Poscondición: informe de asistencia impreso.

Camino estándar

1. El sistema muestra las diferentes opciones.
2. El usuario selecciona la opción de informes.
3. El sistema lista los diferentes informes que puede obtener.
4. El usuario selecciona la opción asistencia de empleados.
5. El sistema solicita los siguientes datos: mes y año de los que se desea obtener informe y la sucursal correspondiente.
6. El usuario ingresa los datos solicitados.
7. El sistema accede al servidor y obtiene de la base de datos la información necesaria.
8. El sistema muestra por empleado todas las fechas en las cuales asistió y por cada una de estas la cantidad de horas trabajadas, además de un total de asistencias del mismo.
9. El usuario selecciona la opción de imprimir.
10. El sistema imprime el informe en papel.

FIN DE CASO DE USO

Caminos alternativos

4.a. El usuario desea informes de asistencia de clientes

4.a.1. El usuario selecciona la opción asistencia de clientes.

4.a.2. El sistema solicita mes, año y sucursal.

4.a.3. El usuario ingresa los datos.

4.a.4. El sistema accede al servidor web y consulta la base de datos.

4.a.5. El sistema muestra por clase, dictada en la sucursal indicada, la cantidad de clientes inscriptos, asistencia promedio diaria y el porcentaje de asistencia correspondiente a cada una.

4.a.6. Retorna al paso 9 del camino estándar.

9.a. El usuario selecciona la opción guardar informe en el ordenador.

9.a.1. El usuario selecciona la opción guardar el informe en el ordenador.

9.a.2. El sistema genera el informe en formato PDF.

9.a.3. El sistema solicita el directorio en donde almacenar.

9.a.4. El usuario selecciona el directorio.

9.a.5. El sistema guarda el archivo en el directorio.

FIN DE CASO DE USO.

- TARJETAS CLASE-RESPONSABILIDAD-COLABORACIÓN (CRC)

SISTEMA	
Propósito: Representa a la clase principal que gestiona el sistema	
Responsabilidad	Colaboración
Conoce información sobre los empleados.	Empleado

EMPLEADO	
Propósito: Representa a un empleado del gimnasio	
Responsabilidad	Colaboración
<p>Almacenar información sobre un empleado.</p> <p>Conocer información acerca de sus asistencias al trabajo.</p>	Asistencia

ASISTENCIA	
Propósito: Representa a la asistencia de una jornada laboral de un empleado.	
Responsabilidad	Colaboración
<p>Almacenar información sobre el cumplimiento horario de los empleados en su labor.</p>	

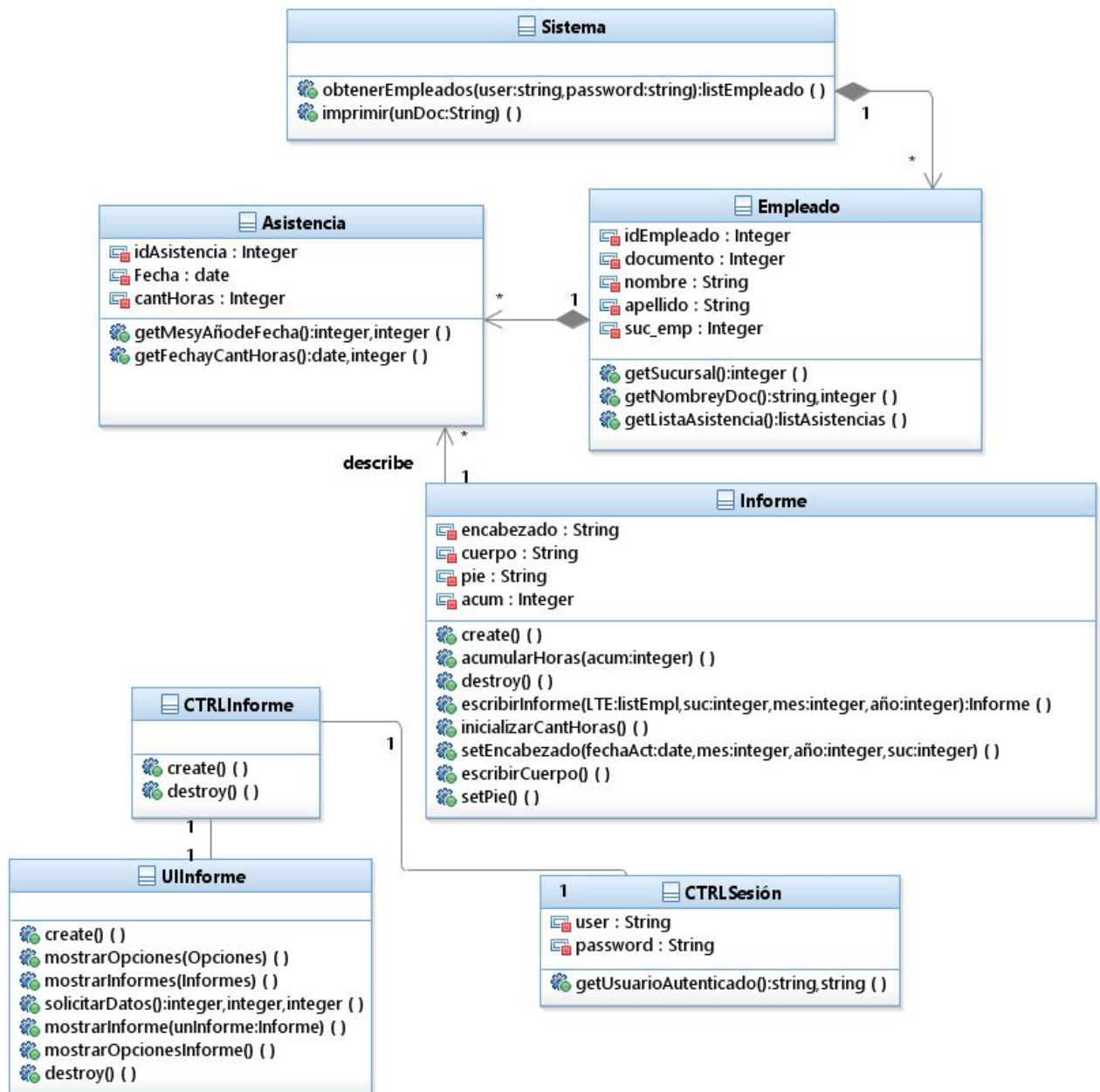
INFORME	
Propósito: Representa a la plantilla que describe un informe.	
Responsabilidad	Colaboración
<p>Describe información de las asistencias de los empleados.</p> <p>Conoce las asistencias de cada empleado.</p>	Asistencia

UInforme	
Propósito: Representa la interfaz entre el actor y el Sistema para generar informe de asistencias.	
Responsabilidad	Colaboración
<p>Muestra opciones en un formato adecuado.</p> <p>Muestra los informes en un formato adecuado.</p> <p>Solicita y recibe confirmación del usuario.</p> <p>Solicita y recibe datos del usuario para generar un informe.</p>	<p>CTRLInforme</p>

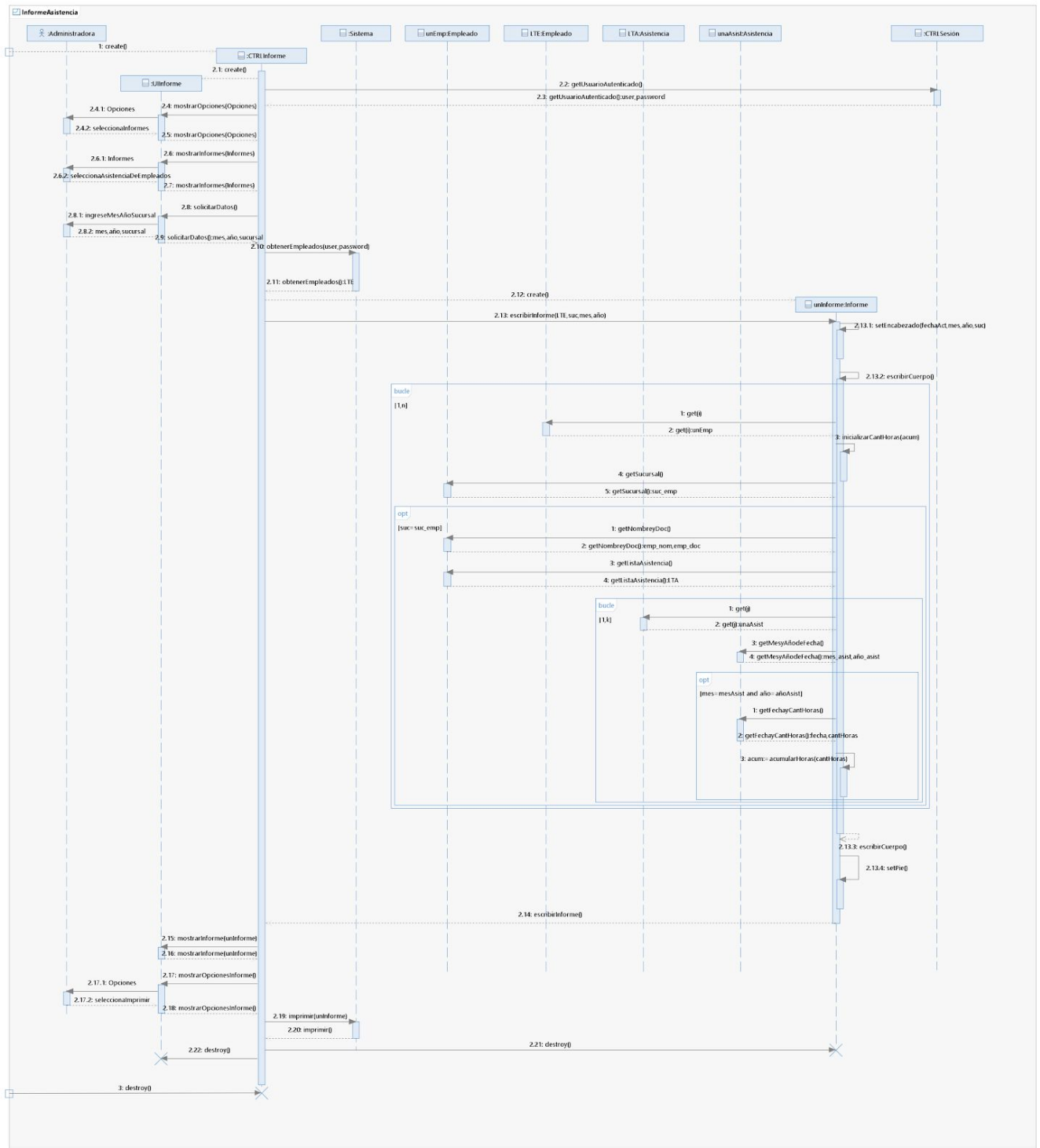
CTRLInforme	
Propósito: Representa al controlador del CU que envía las acciones a ser realizadas al sistema	
Responsabilidad	Colaboración
<p>Crear la UI.</p> <p>Enviar mensajes al sistema sobre las acciones que fueron seleccionadas por el actor.</p> <p>Recibir los resultados de las operaciones realizadas por el sistema</p>	<p>UInforme</p> <p>Sistema</p>

CTRLSesion	
Propósito: Representa al controlador de sesiones	
Responsabilidad	Colaboración
Permite conocer el usuario autenticado en cada momento.	CTRLInforme

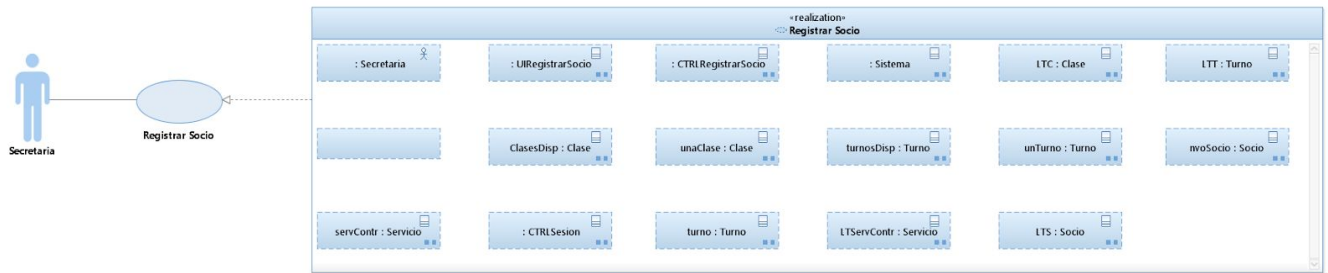
- DIAGRAMA DE CLASES



• DIAGRAMA DE SECUENCIA



REALIZACION DE CASO DE USO: Registrar socio



- DESCRIPCIÓN DE CASO DE USO

Descripción de CU: Registrar nuevo socio.

Actor primario: Secretaria.

Actor secundario: Servidor remoto de base de datos.

Precondición: Asistente autenticada y servidor web disponible.

Postcondición: Nuevo socio registrado.

Camino estándar:

1. El usuario selecciona la opción de registrar socio.
2. El sistema muestra un formulario para completar los datos del socio.
3. El usuario ingresa los siguientes datos solicitados por el sistema: nombre, apellido, dni, telefono, email, domicilio, sexo, foto y antecedentes médicos.
4. El sistema accede al servidor remoto, obtiene las clases o actividades y muestra por pantalla aquellas disponibles llevadas a cabo en la sucursal.
5. El usuario selecciona una clase para inscribir al socio, distinta a complemento.
6. El sistema muestra los turnos disponibles correspondientes a la clase elegida.
7. El usuario selecciona un turno y selecciona guardar.
8. El sistema solicita confirmación.
9. El usuario confirma la operación.
10. El sistema guarda el nuevo socio.

FIN DE CASO DE USO

Camino alternativos

4. a. No hay clases disponibles.

4.a.1. El sistema muestra solo la actividad de complemento.

4.a.2. Sigue en camino alternativo 5.a

5.a. El cliente desea realizar actividad de complemento.

5.a.1. El usuario selecciona la opción de complemento y confirma la operación.

5.a.2. Retorna al paso 8 del camino principal.

7.a. Cancela la operación.

7.a.1. El usuario selecciona cancelar.

7.a.2. El sistema solicita confirmación.

7.a.3. El usuario confirma la operación.

FIN DE CASO DE USO

- TARJETAS CLASE-RESPONSABILIDAD-COLABORACIÓN (CRC)

SISTEMA	
Propósito: Representa a la clase principal que gestiona el sistema.	
Responsabilidad	Colaboración
<p>Conocer a los socios del gimnasio.</p> <p>Conocer las clases que se dictan en el gimnasio.</p>	<p>Socio</p> <p>Clase</p>

CLASE	
Propósito: Representa una clase o actividad del gimnasio.	
Responsabilidad	Colaboración
<p>Almacenar información referida a la actividad o clase.</p> <p>Conocer los turnos en los cuales se realizan las mismas.</p>	Turno

TURNO	
Propósito: Representa el dictado de una clase en un horario y días determinados de la semana.	
Responsabilidad	Colaboración
Almacenar información referida a un dictado específico con sus correspondientes responsables y cupos.	

SOCIO	
Propósito: Representa a un cliente que asiste a clases o actividades del gimnasio.	
Responsabilidad	Colaboración
<p>Almacenar información referida al cliente.</p> <p>Conocer los servicios que contrata.</p>	Servicio

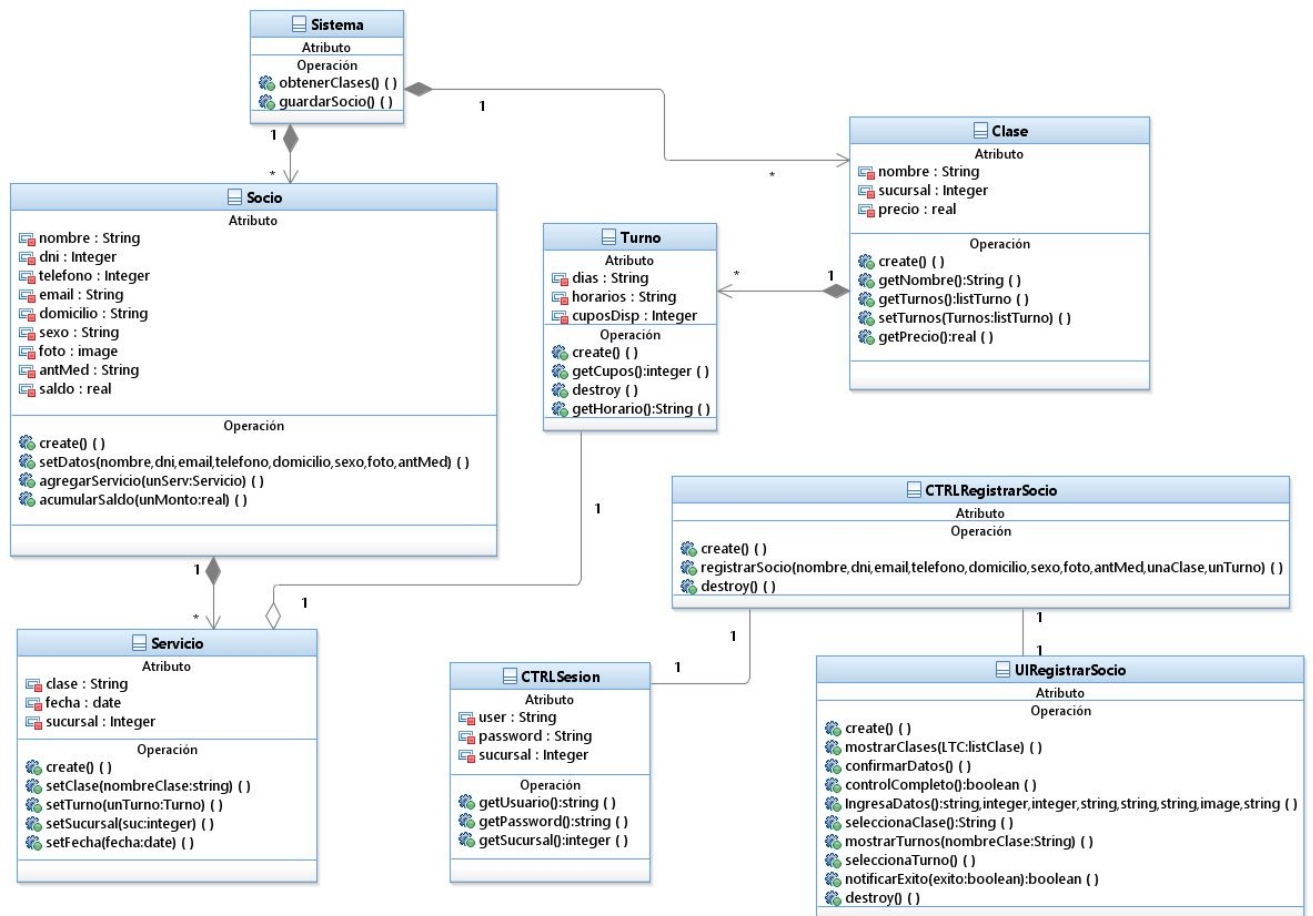
SERVICIO	
Propósito: Representa la inscripción de un cliente en una clase.	
Responsabilidad	Colaboración
<p>Almacenar la inscripción de un cliente a una clase u actividad llevada a cabo en el gimnasio.</p> <p>Conocer el turno de la clase contratada por el cliente.</p>	Turno

UIRegistrarSocio	
Propósito: Representa a la interfaz entre el actor y el sistema para realizar el registro de un nuevo socio	
Responsabilidad	Colaboración
<p>Recibir mensajes de los datos del socio para la inscripción.</p> <p>Recibir la selección de clase y turno.</p> <p>Mostrar las clases con sus turnos disponibles en un formato adecuado.</p> <p>Recibir y enviar datos de confirmación.</p>	CTRLRegistrarSocio

CTRLRegistrarSocio	
Propósito: Representa al controlador del CU que envía las acciones a ser realizadas al sistema	
Responsabilidad	Colaboración
<p>Crear la UI.</p> <p>Enviar mensajes al sistema sobre las acciones que fueron seleccionadas por el actor.</p> <p>Recibir los resultados de las operaciones realizadas por el sistema.</p>	<p>UIRegistrarSocio</p> <p>Sistema</p>

CTRLSesion	
Propósito: Representa al controlador de sesiones.	
Responsabilidad	Colaboración
<p>Permite conocer el usuario autenticado en cada momento y su sucursal.</p>	<p>CTRLRegistrarSocio</p>

• DIAGRAMA DE CLASES



- DIAGRAMA DE SECUENCIA

- IMPLEMENTACION

Lenguaje elegido: Python

Librería utilizada para implementar UI: PyQt

class sistema:

```
def __init__(self):
    self.__socios=list()

def obtenerClases(self,userName, password,sucursal):
    return LTClases

def guardarSocio(self, socio):
    self.__socios.append(socio)
    return False
```

class clase:

```
def __init__(self,nombre,precio, turnos):
    self.__nombre=nombre
    self.__precio=precio
    self.__turnos=turnos

def getTurnos(self):
    return self.__turnos

def getNombre(self):
    return self.__nombre

def getPrecio(self):
    return self.__precio

def setTurnos(self,turnos):
    self.__turnos=turnos
```

class turno:

```
def __init__(self,dia, hora, cupos):
    self.__dia=dia
    self.__hora=hora
    self.__cupos=cupos

def getCupos(self):
    return self.__cupos

def getHorario(self):
    return (self.__dia+' '+self.__hora)
```

class socio:

```
def __init__(self):
    self.__nombre=None
    self.__dni=None
    self.__email=None
    self.__telefono=None
    self.__domicilio=None
    self.__sexo=None
    self.__foto=None
    self.__antMed=None
    self.__saldo=0
```

```
self.__servicios=list()

def setDatos(self, nombre, dni, email, tel, domicilio, sexo, foto, antMed):
    self.__nombre=nombre
    self.__dni=dni
    self.__email=email
    self.__telefono=tel
    self.__domicilio=domicilio
    self.__sexo=sexo
    self.__foto=foto
    self.__antMed=antMed

def acumularSaldo(self, monto):
    self.__saldo=self.__saldo+monto

def agregarServicio(self, clase, turno, sucursal):
    servContratado=servicio()
    servContratado.setClase(clase)
    servContratado.setTurno(turno)
    servContratado.setSucursal(sucursal)
    servContratado.setFecha(time.strftime("%d/%m/%y"))
    self.__servicios.append(servContratado)

class servicio:
    def __init__(self):
        self.__clase=None
        self.__turno=None
        self.__sucursal=None
        self.__fecha=None

    def setClase(self, clase):
        self.__clase=clase

    def setTurno(self, turno):
        self.__turno=turno

    def setSucursal(self, suc):
        self.__sucursal=suc

    def setFecha(self, fecha):
        self.__fecha=fecha

class CTRLsesion:

    def __init__(self):
        self.__usuario='secretaria'
        self.__contraseña='password'
        self.__sucursal=1

    def getUsuario(self):
        return self.__usuario

    def getPassword(self):
        return self.__contraseña

    def getSucursal(self):
        return self.__sucursal

class CTRLregistrarSocio:
    def create(self):
        pass
```

```
def destroy(self):
    pass

def ejecutar(self):
    app = QApplication(sys.argv)
    UI=UIregistrarSocio()
    UI.show()
    userName=CTRLsesion().getUsuario()
    password=CTRLsesion().getPassword()
    suc=CTRLsesion().getSucursal()
    LTClases=sistema().obtenerClases(userName,password,suc)
    clasesDisp=list()
    for i in range(len(LTClases)):
        unaClase=LTClases[i]
        if unaClase.getNombre()!='Complemento':
            LTTurno=unaClase.getTurnos()
            turnosDisp=list()
            for j in range(len(LTTurno)):
                unTurno=LTTurno[j]
                cuposTurno=unTurno.getCupos()
                if cuposTurno>0:
                    turnosDisp.append(unTurno)
            if len(turnosDisp)>0:
                unaClase.setTurnos(turnosDisp)
                clasesDisp.append(unaClase)
            del turnosDisp
        else:
            clasesDisp.append(unaClase)
    UI.mostrarClases(clasesDisp)
    app.exec_()

def registrarsocio(self, datos):
    nuevoSocio=socio()
    #nombre, dni, email, telefono, sexo, foto, antMedicos
    nuevoSocio.setDatos(datos[0],datos[1],datos[2],datos[3],datos[4],datos[5],datos[6],datos[7])
    unaClase=datos[len(datos)-2]
    monto=unaClase.getPrecio()
    nuevoSocio.acumularSaldo(monto)
    unTurno=datos[len(datos)-1]

    suc=CTRLsesion().getSucursal()
    nuevoSocio.agregarServicio(unaClase.getNombre(), unTurno, suc)

    exito=sistema().guardarSocio(nuevoSocio)

    return exito
class UIregistrarSocio(QMainWindow, CTRLregistrarSocio):
    def __init__(self):
        QMainWindow.__init__(self)
        uic.loadUi("registrarSocio.ui", self)
        self.clases=list()
        self.actividad.currentTextChanged.connect(self.seleccionarClase)
        self.botonGuardar.clicked.connect(self.confirmarDatos)
        self.botonCancelar.clicked.connect(self.cancelarOp)
        self.botonFoto.clicked.connect(self.capturarFoto)

def controlCompleto(self):
    if self.nombre.text()!=" and self.dni.text()!=":
        self.nombre.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.dni.setStyleSheet("background-color: rgb(255, 255, 255);")
```

```

        return True
    else:
        if self.nombre.text()=="":
            self.nombre.setStyleSheet("background-color: rgb(255, 255, 127);")
        if len(self.dni.text())<8:
            self.dni.setStyleSheet("background-color: rgb(255, 255, 127);")
        return False

def mostrarClases(self,clasesDisp):
    self.clases=clasesDisp
    for unaClase in clasesDisp:
        self.actividad.addItem(unaClase.getNombre())

def seleccionarClase(self):
    if self.actividad.currentText()!='Complemento':
        self.labelTurno.setVisible(True)
        self.turno.setVisible(True)
        self.mostrarTurnos(self.actividad.currentText())
    else:
        self.turno.setVisible(False)
        self.labelTurno.setVisible(False)
        self.turno.clear()

def mostrarTurnos(self,claseSelec):
    self.turno.clear()
    for unaClase in self.clases:
        if unaClase.getNombre()==claseSelec:
            LTTurno=unaClase.getTurnos()
            for turno in LTTurno:
                self.turno.addItem(turno.getHorario())
            break

def confirmarDatos(self):

    if self.controlCompleto():
        confirma=QMessageBox.question(self, "Aviso", "Desea guardar los datos ingresados para el nuevo
socio?",QMessageBox.No|QMessageBox.Yes, QMessageBox.Yes )

        if confirma==QMessageBox.Yes:
            #nombre, dni, email, tel, domicilio, sexo, foto, antMed
            datos=[self.nombre.text(), self.dni.text(), self.email.text(), self.telefono.text(), self.domicilio.text() ] #falta
            antecedentes medicos
            if self.masculino.isChecked():
                datos.append('M')
            else:
                datos.append('F')
            for clase in self.clases:
                if clase.getNombre()==self.actividad.currentText():
                    #print('clase elegida', clase.getNombre())
                    unaClase=clase
                    unTurno=None
                    #print(claseSelec)
                    if self.actividad.currentText()!='Complemento':
                        for turno in clase.getTurnos():
                            if turno.getHorario()==self.turno.currentText():
                                unTurno=turno
                                break
                        break
                    datos.append(self.foto)
                    datos.append(self.antMed.document())
                    datos.append(unaClase)
                    datos.append(unTurno)

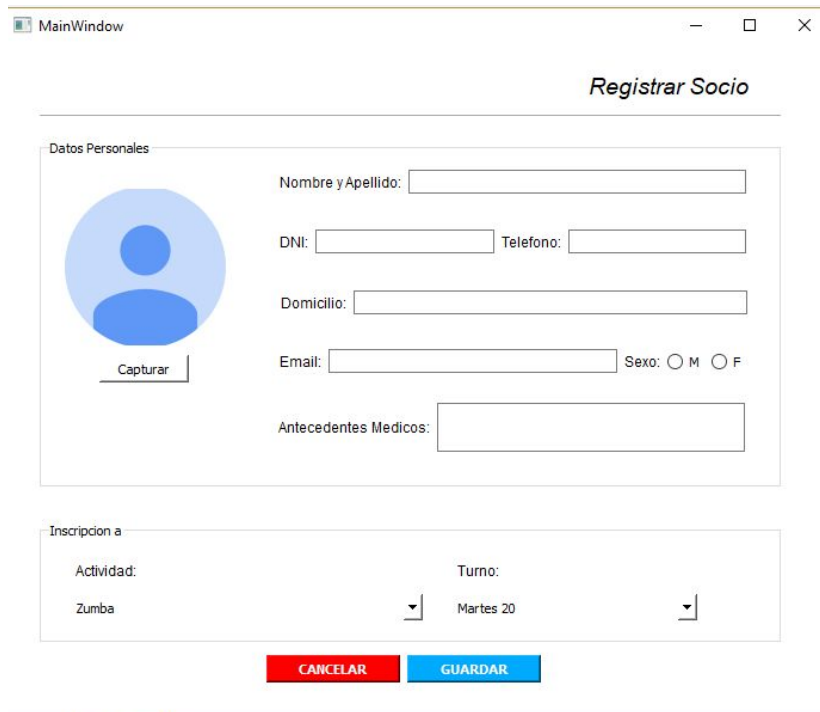
```

```

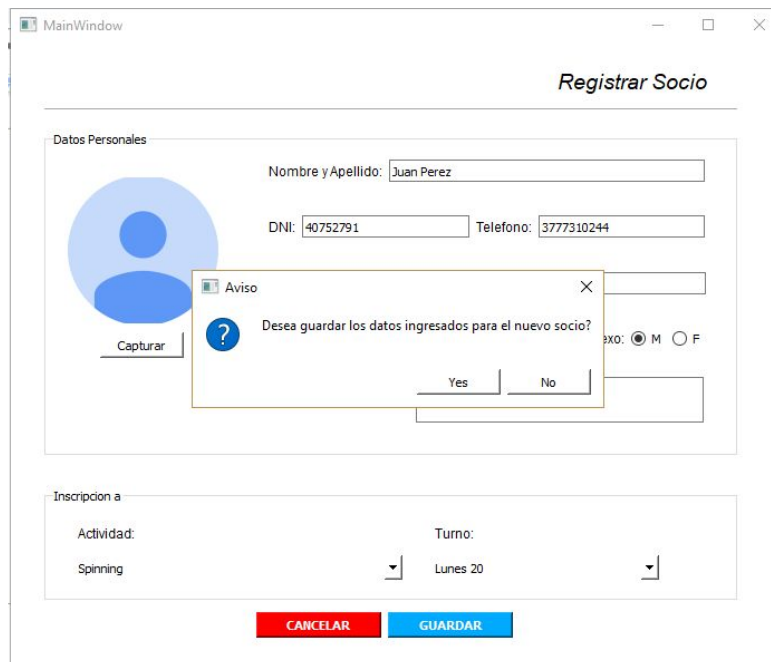
if CTRLregistrarSocio.registrarsocio(self, datos):
    self.notificarExito(True)
else:
    self.notificarExito(False)
    
```

Interfaz gráfica para la implementación

- Formulario de datos:



- Confirmación de operación:



- Notificación de éxito:

The screenshot shows a web application window titled 'MainWindow' with a form titled 'Registrar Socio'. The form is divided into two main sections: 'Datos Personales' and 'Inscripcion a'. In the 'Datos Personales' section, there is a circular profile picture placeholder with a blue silhouette and a 'Capturar' button below it. To the right of the profile picture are input fields for 'Nombre y Apellido' (containing 'Juan Perez'), 'DNI' (containing '40752791'), and 'Telefono' (containing '3777310244'). Below these fields is a 'Sexo' section with radio buttons for 'M' (selected) and 'F'. A modal dialog box titled 'Aviso' is open in the center of the form, displaying the message 'Nuevo socio registrado correctamente' and an 'OK' button. Below the 'Datos Personales' section is the 'Inscripcion a' section, which includes a dropdown menu for 'Actividad' (currently showing 'Spinning') and another dropdown menu for 'Turno' (currently showing 'Lunes 20'). At the bottom of the form are two buttons: 'CANCELAR' (red) and 'GUARDAR' (blue).

- Cancelar operación:

This screenshot shows the same 'Registrar Socio' form as the previous one, but with a different modal dialog box open. The dialog box, titled 'Aviso', now displays a question mark icon and the text 'Desea cancelar la operacion?'. It has two buttons: 'Yes' and 'No'. The background form is slightly dimmed, showing the same fields for 'Nombre y Apellido', 'DNI', 'Telefono', 'Sexo', 'Actividad', and 'Turno', along with the 'Capturar' button and the 'CANCELAR' and 'GUARDAR' buttons at the bottom.