<<NOMBRE DEL CLIENTE>>

**Documento de Especificación de Arquitectura**

Realizado por:

**Gilberto Salinas Linares**

Correo: gsalinas0@misena.edu.co

**Yeimy Lorena Forero**

Correo:ylforero0@misena.edu.co

**Johan Steven Morales**

Correo:jsmorales067@misena.edu.co

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
| 11/07/2016 <<Fecha creación y/o modificación>> | FENIX RI  1. 0<<Versión>> | Gilberto Salinas Linares  Yeimy Lorena Forero  Johan Steven Morales<<Creador o quien solicita ajustes>> | Sistema de información para el seguimiento y control de los estudiantes y sus evoluciones <<Persona que revisa el documento>> | <<Persona que aprueba el documento>> |

Contenido

[1. Documento de Arquitectura de Software 4](#_Toc398014035)

[1.1. Introducción 4](#_Toc398014036)

[1.2. Propósito 4](#_Toc398014037)

[1.3. Alcance 4](#_Toc398014038)

[1.4. Referencias 5](#_Toc398014039)

[1.5. Definiciones acrónimos y abreviaciones 5](#_Toc398014040)

[2. Generalidades del Proyecto 5](#_Toc398014041)

[2.1. Problema a Resolver 5](#_Toc398014042)

[2.2. Descripción General del Sistema a Desarrollar 5](#_Toc398014043)

[2.3. Identificación de los Stakeholders y sus responsabilidades 5](#_Toc398014044)

[3. Vistas de la arquitectura 6](#_Toc398014045)

[3.1. Vista de Casos de Uso 6](#_Toc398014046)

[3.2. Vista de Procesos 6](#_Toc398014047)

[3.3. Vista Lógica 6](#_Toc398014048)

[3.4. Vista de Implementación 6](#_Toc398014049)

[3.5. Vista de Despliegue 6](#_Toc398014050)

[4. Arquitectura en capas 6](#_Toc398014051)

[5. Vista de Datos 7](#_Toc398014052)

[5.1. Modelo Relacional 7](#_Toc398014053)

[6. Definición de Interfaces de Usuario 7](#_Toc398014054)

[7. Características Generales de Calidad 7](#_Toc398014055)

[7.1. Tamaño y performance 7](#_Toc398014056)

[7.2. Calidad 7](#_Toc398014057)

[7.3. Usabilidad 7](#_Toc398014058)

[7.4. Eficiencia 7](#_Toc398014059)

[7.5. Seguridad 7](#_Toc398014060)

[7.6. Confiabilidad 7](#_Toc398014061)

[7.7. Mantenimiento 7](#_Toc398014062)

[7.8. Estándares 7](#_Toc398014063)

# Documento de Arquitectura de Software

## Introducción

Este documento evidencia la arquitectura y funcionalidades del sistema de información FÉNIX RI, con sus respectivos diagramas, descripción del documento y referencias al momento de redactar el actual documento.

## Propósito

Este documento proporciona una descripción comprensiva arquitectónica del sistema, usando un número finito de vistas diferentes para representar los distintos aspectos que se requieren para capturar y transportar las decisiones significativas que han sido hechas sobre el sistema.

## Alcance

Este documento de arquitectura tiene como alcance dar una visión global de las vistas de despliegue y de Implementación de la arquitectura de software del sistema de Información FENIX RI, el cual está pensado para la gestión deportiva y administrativa del club y en un futuro implementar el software a la academia colombo coreana.

## Referencias

1. IEEE 830-1998 ST.
2. Documento de análisis del sistema.
3. Documento de especificación de arquitectura

## Definiciones acrónimos y abreviaciones

WTF: World Taekwondo Federation

Fedeco: Federación colombo coreana

LTKDB: Liga de taekwondo de Bogotá

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

**ESCENARIO**: Los stakeholders desempeñan su función de instructores en la academia coreana y en los parques, para que sus entrenamientos sean de esparcimiento e interesantes.

# Generalidades del Proyecto

## Problema a Resolver

Se pretende optimizar el seguimiento de los estudiantes en el entorno de las evoluciones deportivas, además se tendrá un control más exacto de las mensualidades de cada uno de ellos, también se quiere tener un registro de los estudiantes en cuanto a cantidad se refiere y cuántos de estos están trasladados o retirados.

## Descripción General del Sistema a Desarrollar

Será un sistema de información capaz de monitorear a los estudiantes en los siguientes aspectos:

Deportivamente: el instructor llevará las evoluciones deportivas de cada estudiante y así acertar en las fechas de exámenes y control del tiempo que el estudiante lleva en el club.

Académicamente: el instructor podrá compartir documentación importante que el estudiante requiere para avanzar en cuanto conocimiento y así poderse capacitar más para los exámenes de ascenso de grado.

Administrativamente: llevando un conteo del personal que se encuentra actualmente en el club y los estudiantes trasladado y retirado, además se llevará un historial de pagos, para evitar moras y posibles “fraudes de pagos”.

Eventos: mediante un calendario de actividades los estudiantes podrán evidenciar las fechas que se han propuesto por parte de instructor para entrenamientos especiales, torneos, exámenes de ascenso de grado e intercambios culturales con otros clubes.

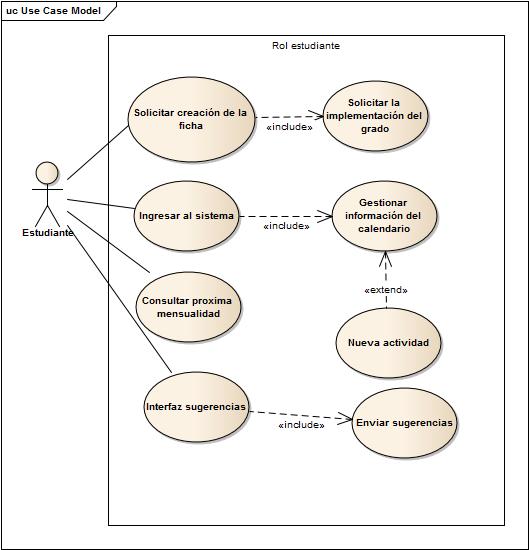
De esta forma el sistema de información FENIX RI pretende atender las necesidades del club y en un futuro ser implementado en la academia colombo coreana.

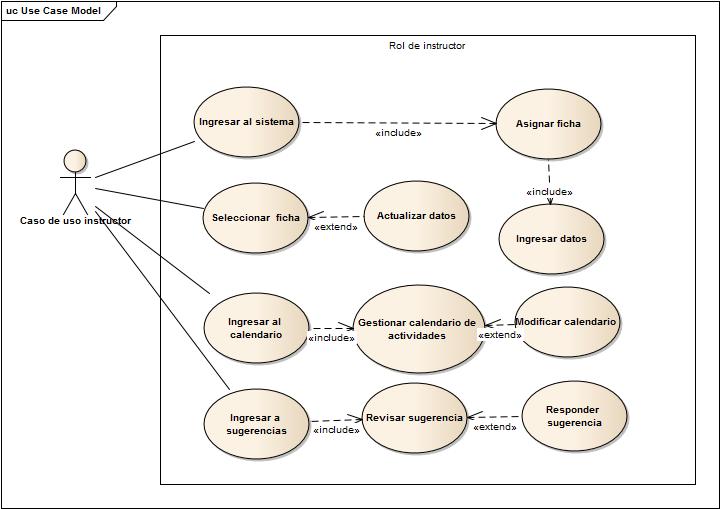
## Identificación de los Stakeholders y sus responsabilidades

|  |  |  |  |
| --- | --- | --- | --- |
| **STAKEHOLDER** | **DESCRIPCIÓN** | **ESCENARIO** | **Caso de Uso** |
| Instructor | Es el usuario encargado de administrar el sistema de información y tiene la responsabilidad de ingresar nuevos miembros a la base de datos del sistema. | * Escenario de   negocios   * Escenario de   Diseño  Escenario deportivo | * CU\_Ingresar al sistema * CU\_Gestionar registros de los estudiantes * CU\_Ingresar datos * CU\_Asignar grados de los estudiantes ingresados * CU\_Seleccionar fichas * CU\_Actualizar fichas de los estudiantes * CU\_Ingresar al calendario * CU\_Gestionar calendario de actividades * CU\_Modificar calendario * CU\_Ingresar a sugerencias * CU\_Revisar sugerencias * CU\_Responder sugerencias |
| Estudiante | Es la persona que  interactúa con el  sistema de información para consultar las fechas de los eventos, documentos de aprendizaje y pagos que realizar. | * Escenario academico * Escenario de   Diseño  Escenario deportivo | * CU\_Gestionar creación de la   Ficha técnica   * CU\_Gestionar Implementación del grado * CU\_Ingresar al sistema * CU\_Información del calendario * CU\_Revisar nueva actividad * CU\_Consultar próxima mensualidad * CU\_Interfaz sugerencias * CU\_Enviar sugerencias |

# Vistas de la arquitectura

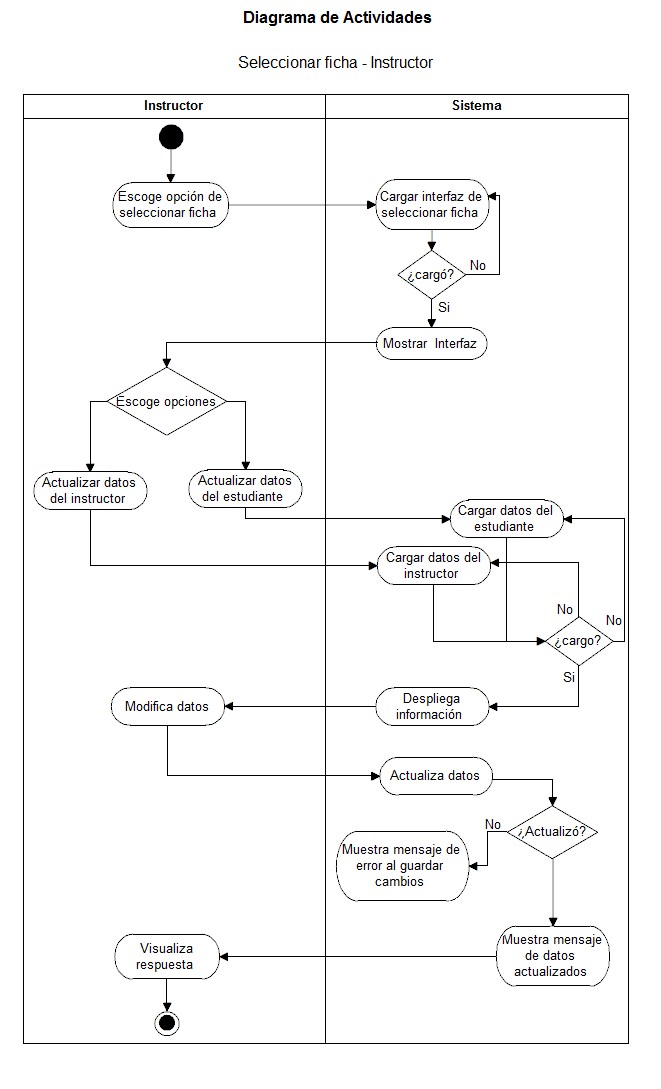
## Vista de Casos de Uso

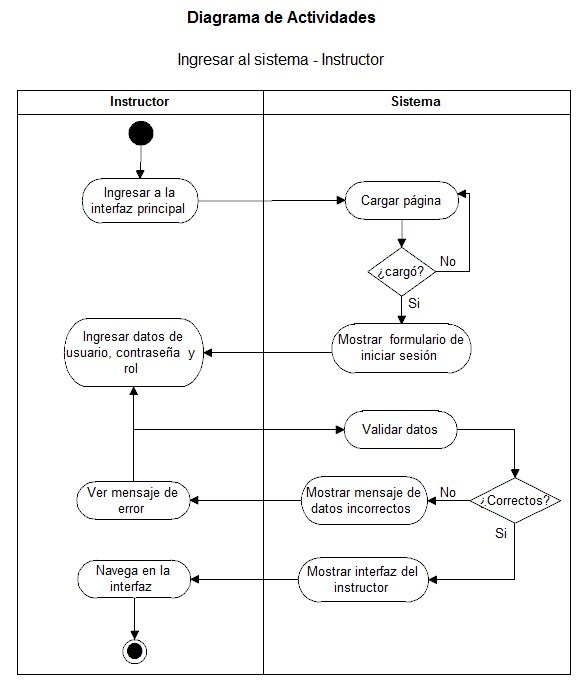


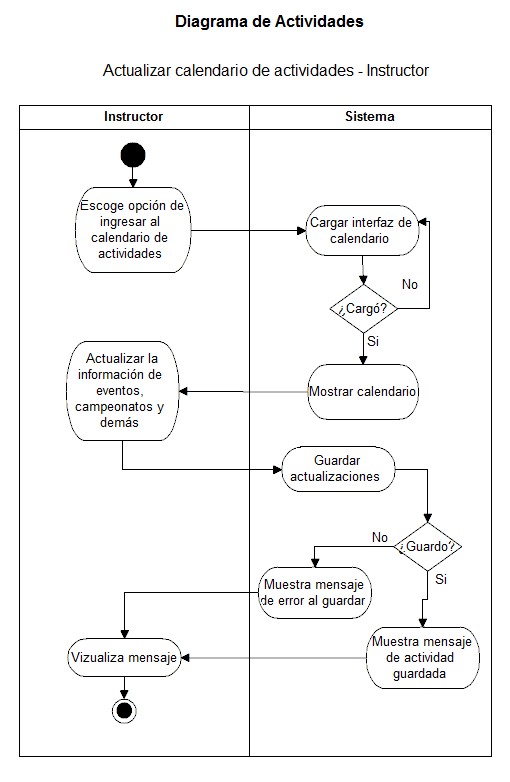


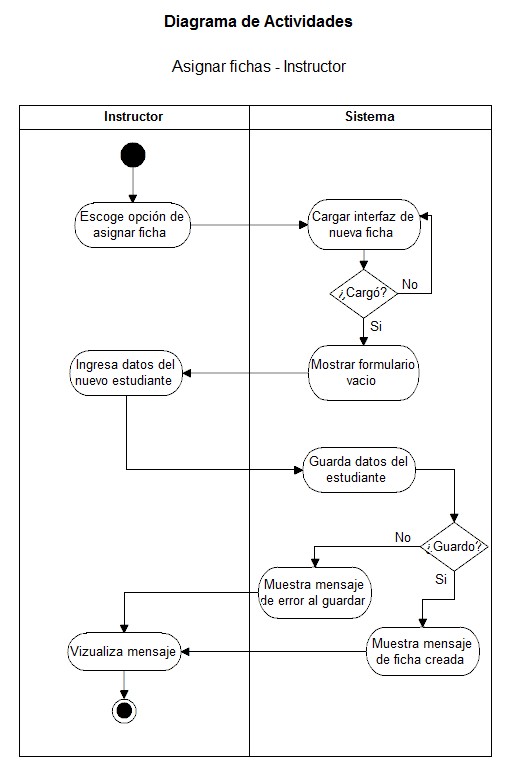
## Vista de Procesos

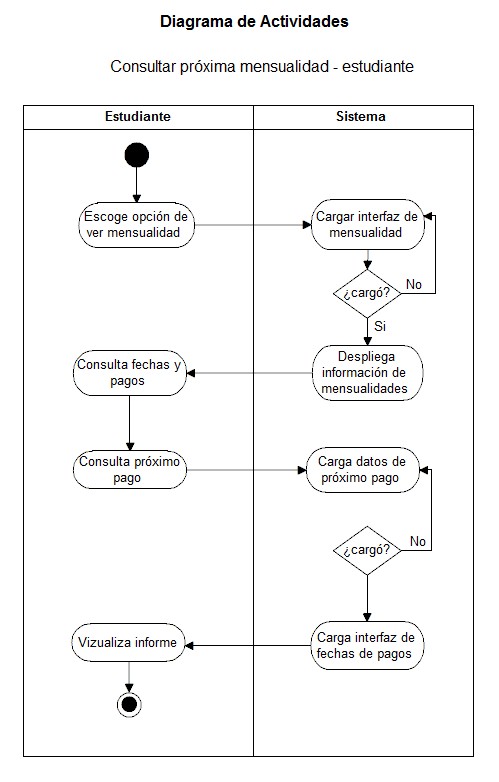
### Diagrama de Actividades

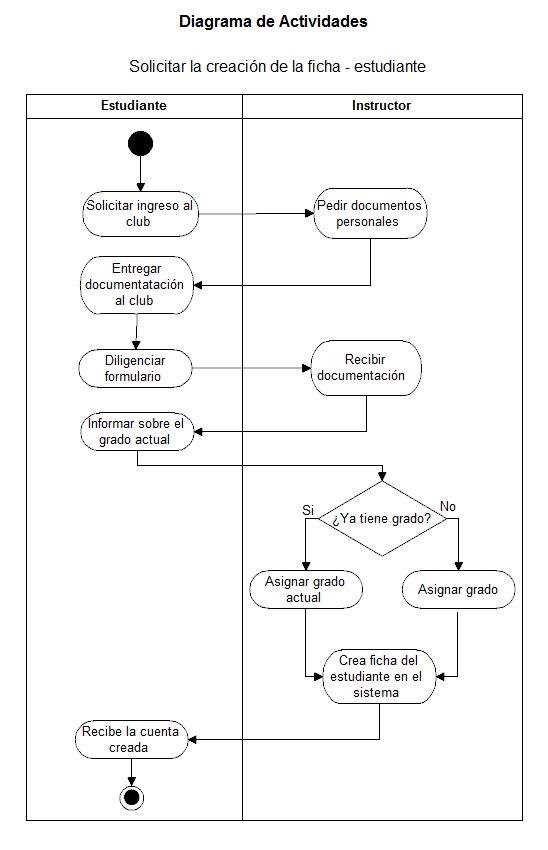




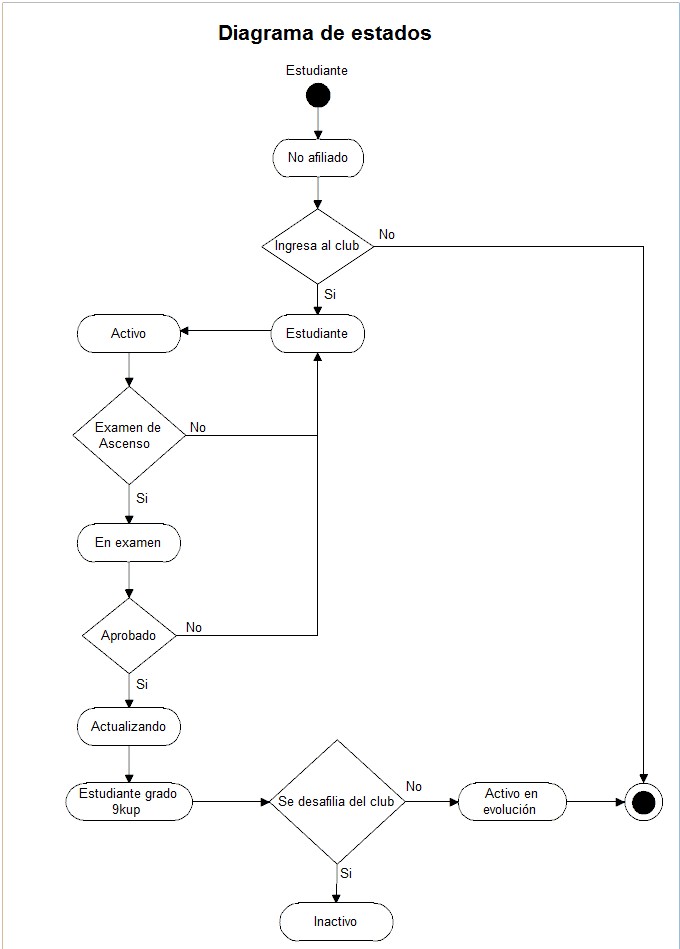


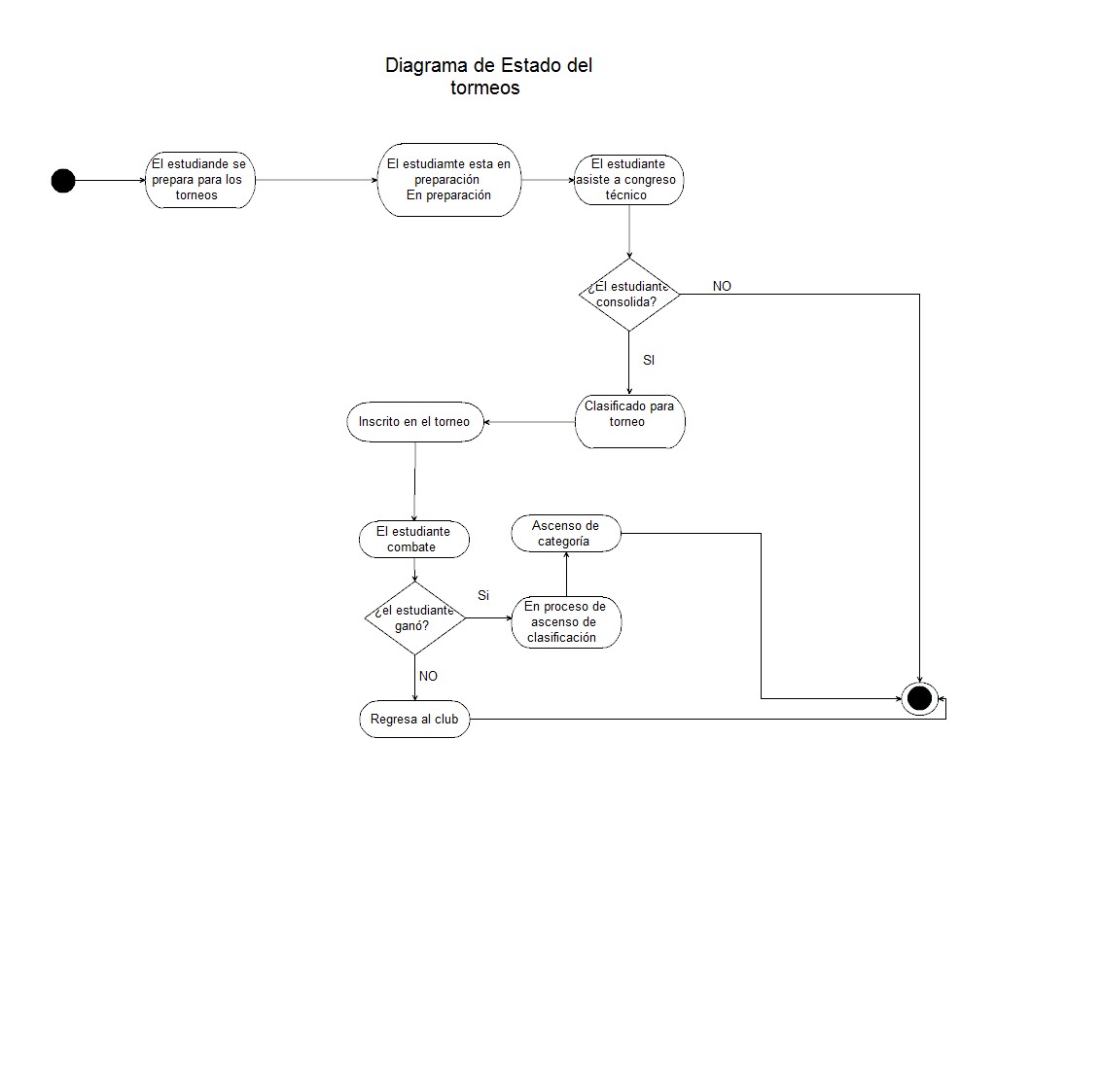






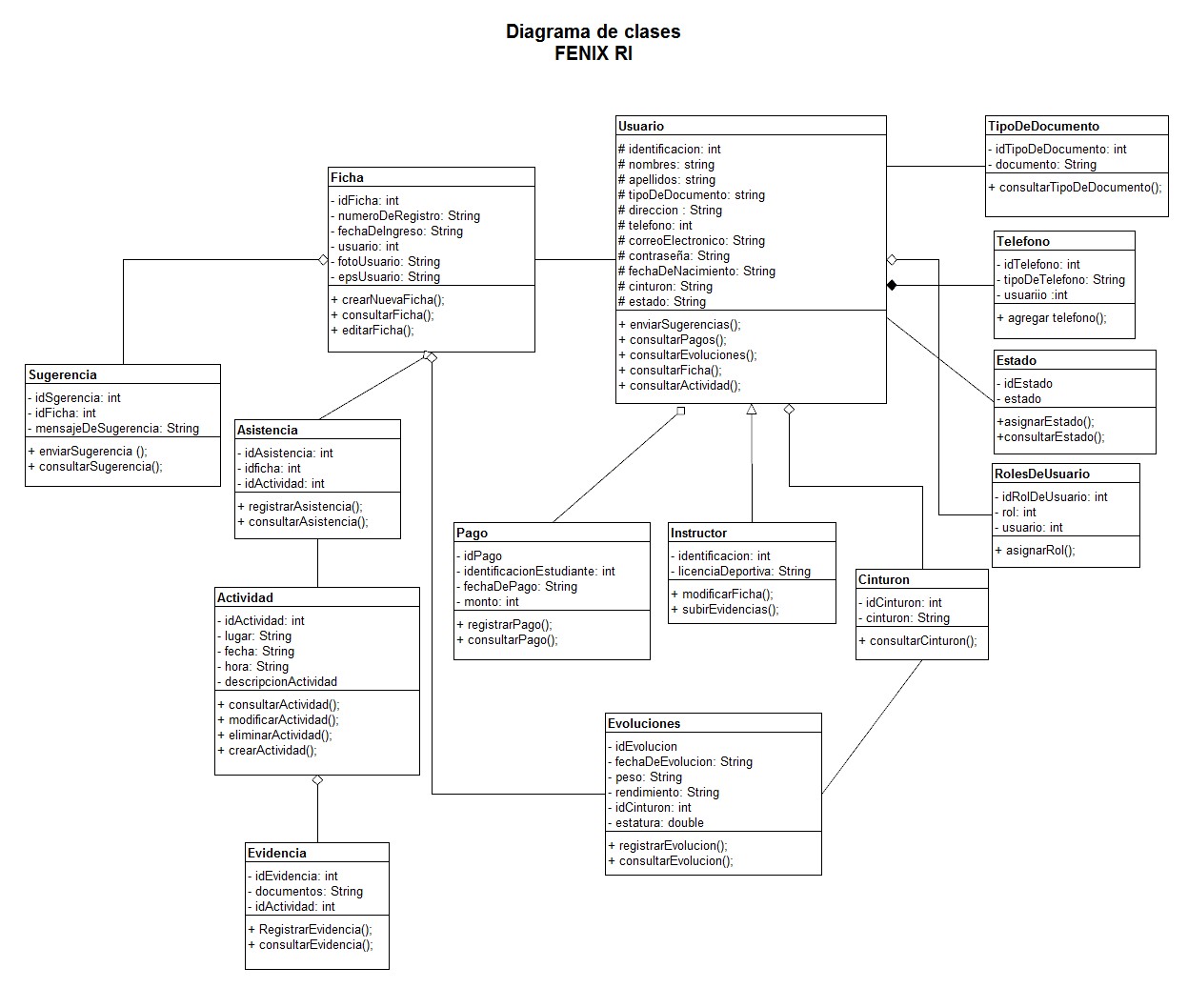
### Diagrama de estados



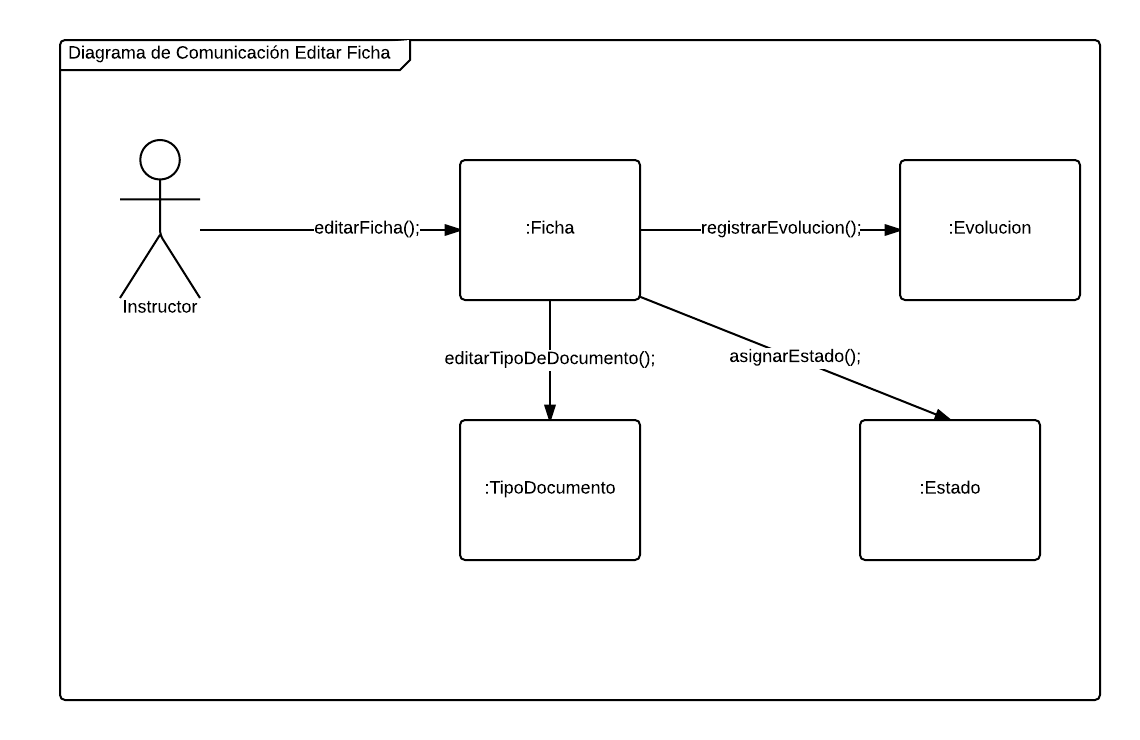


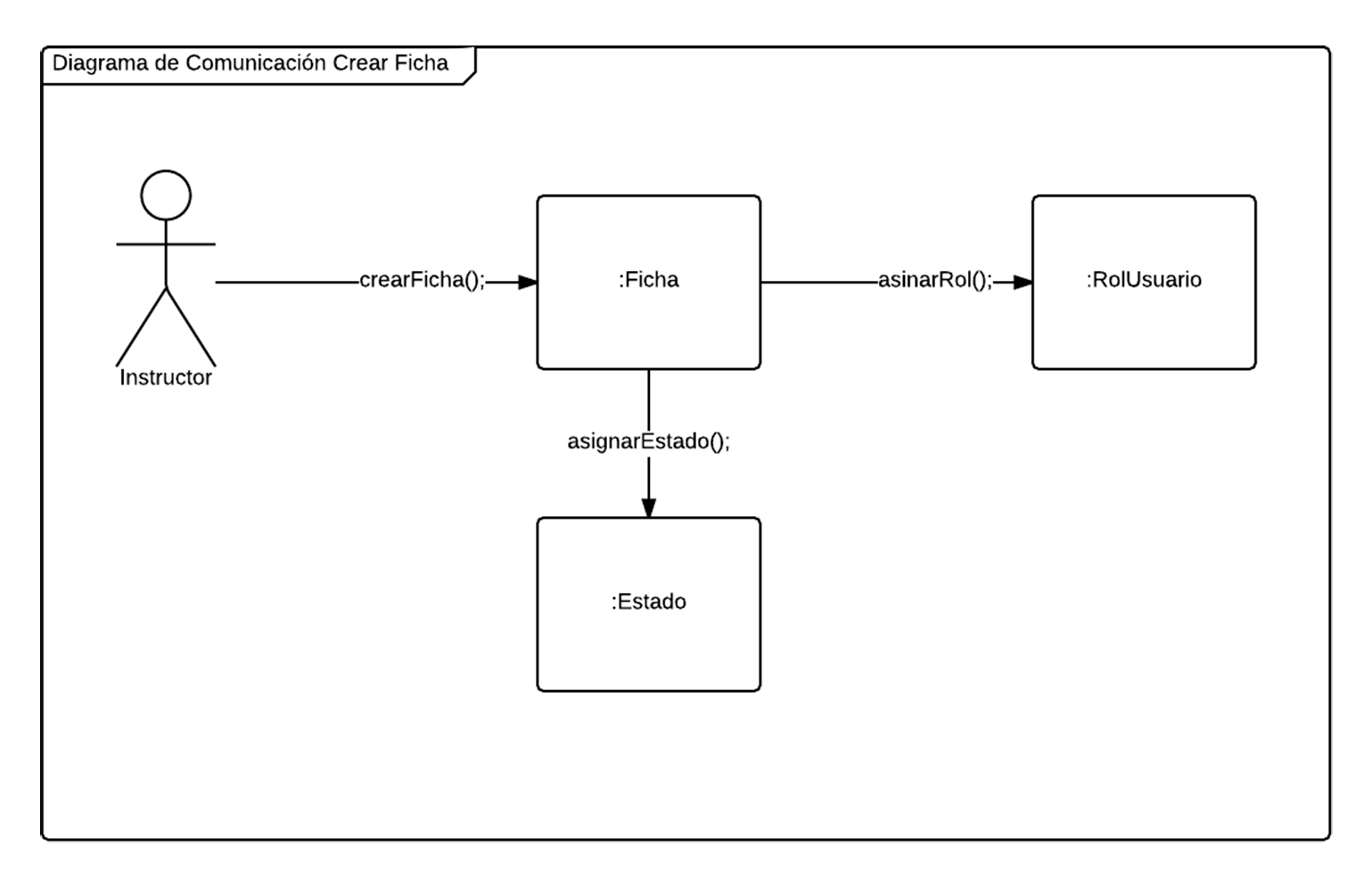
## Vista Lógica

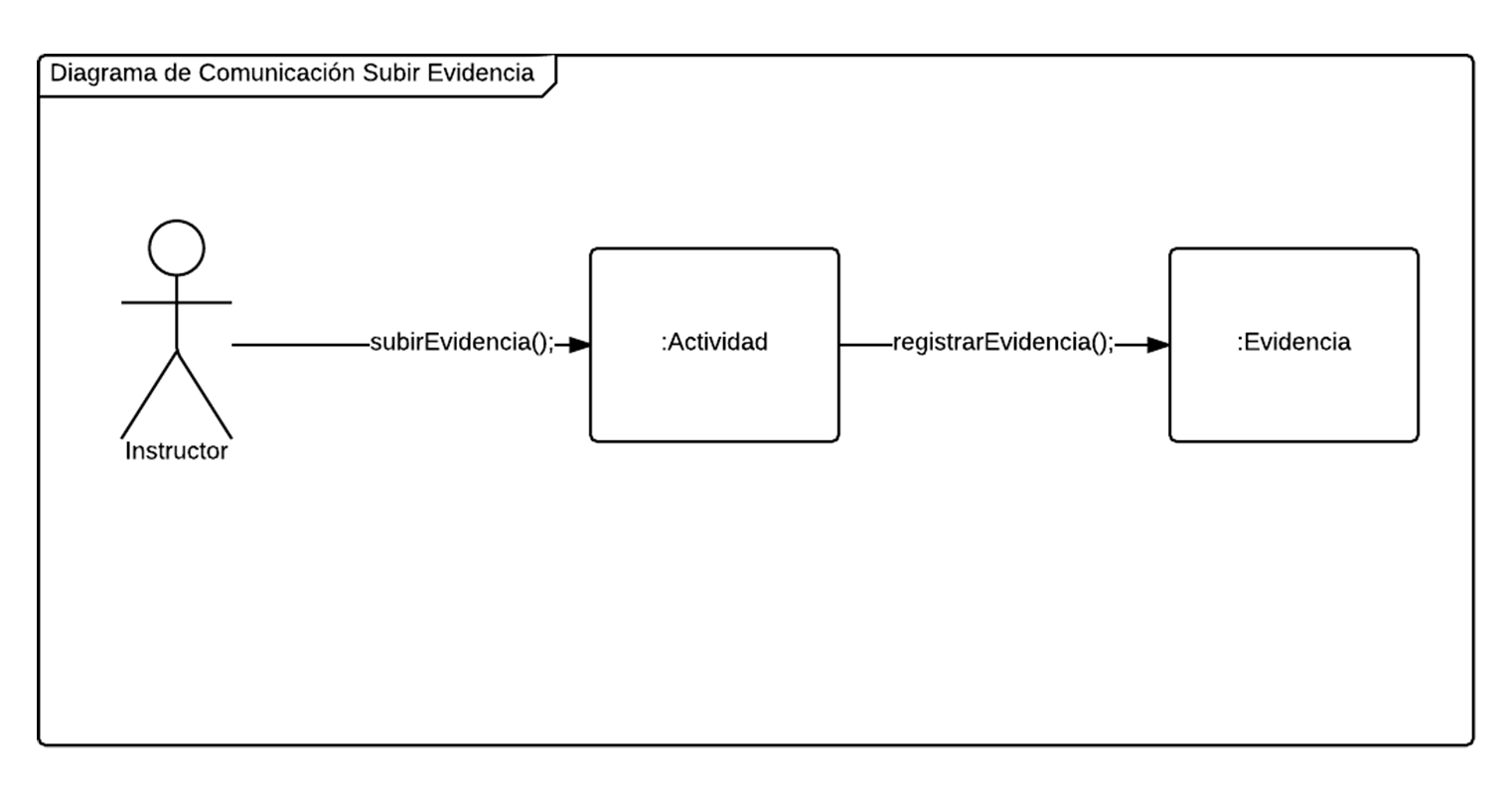
### Diagramas – Clases

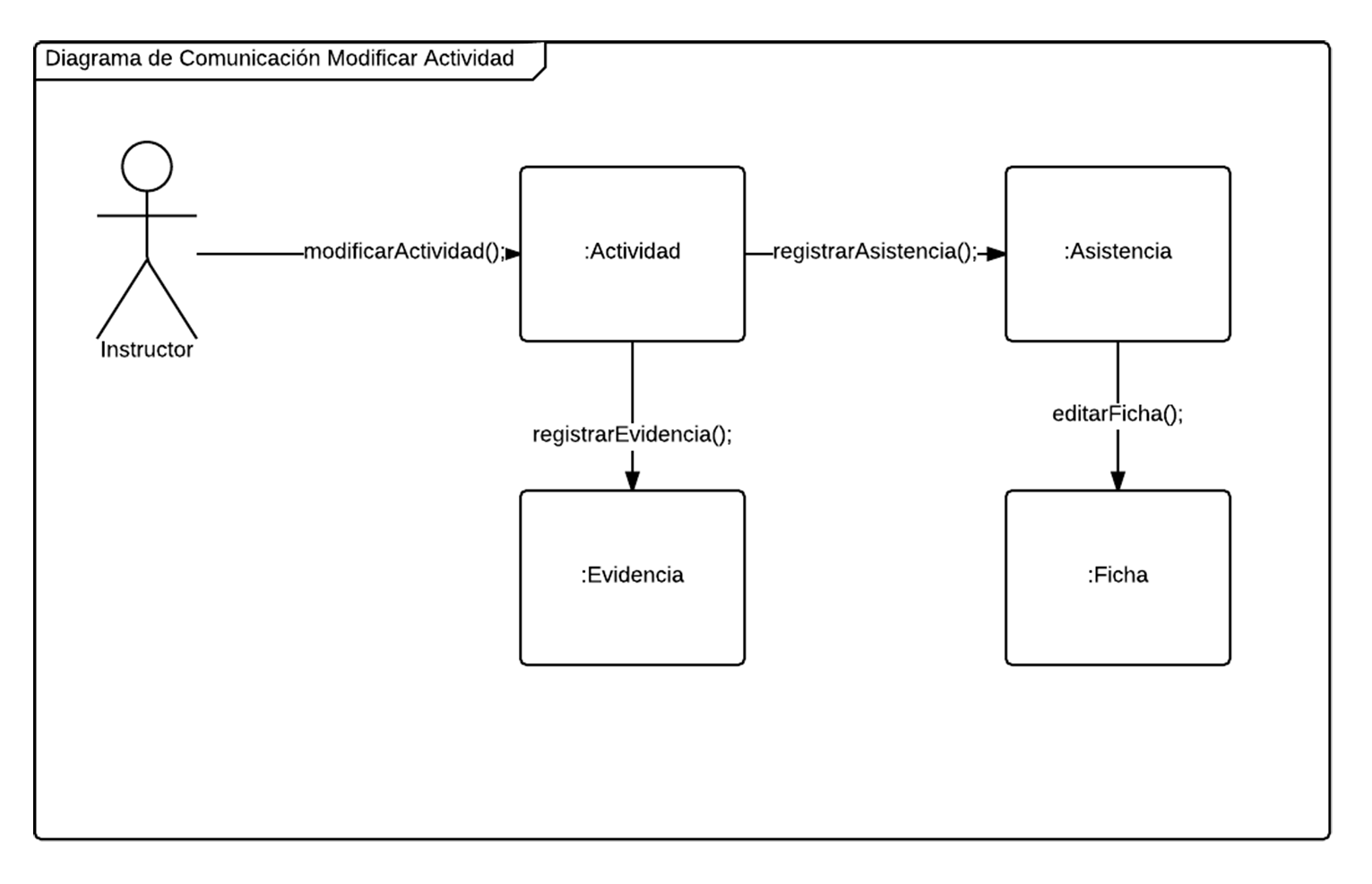


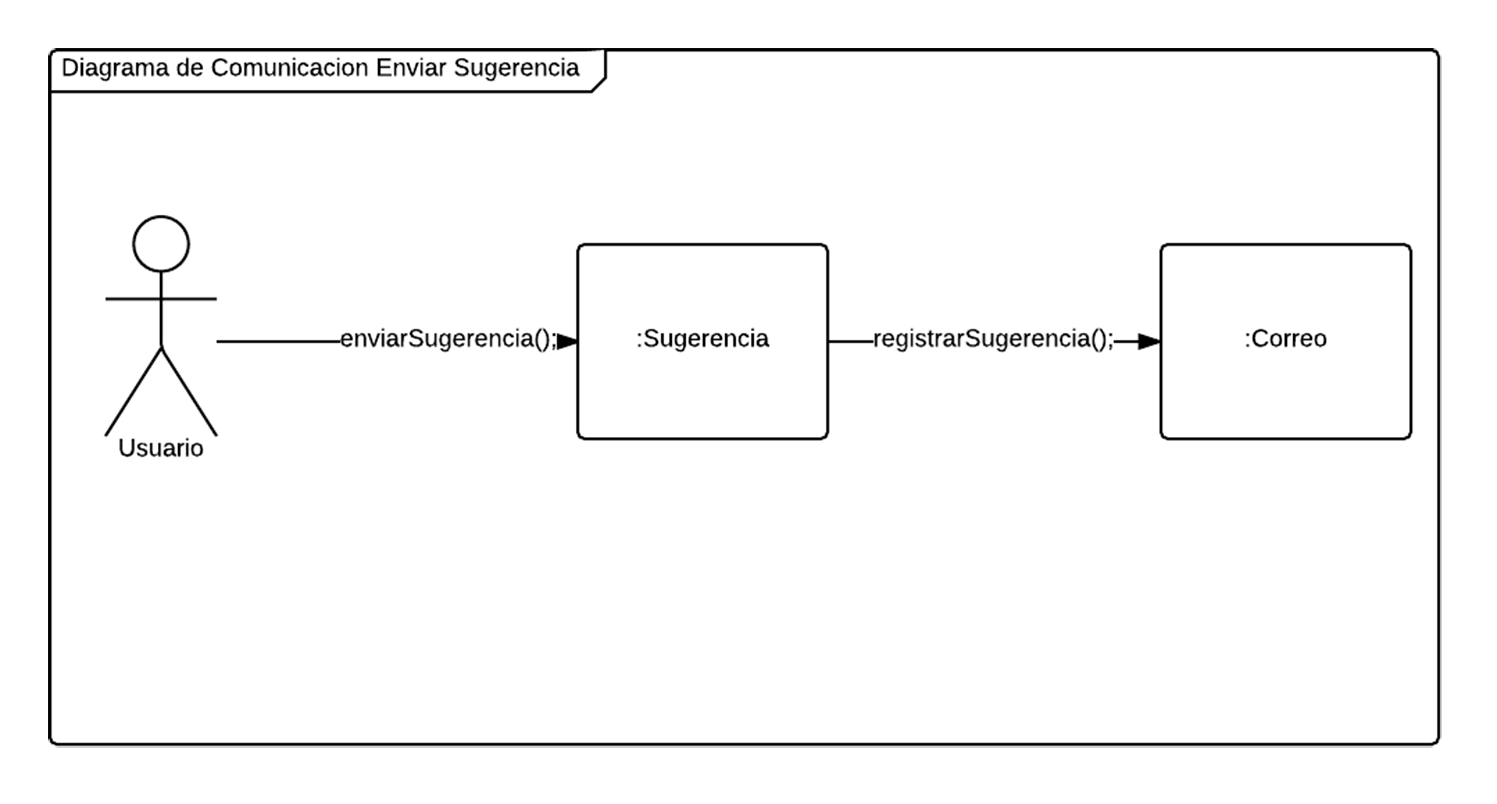
### Diagrama de comunicación

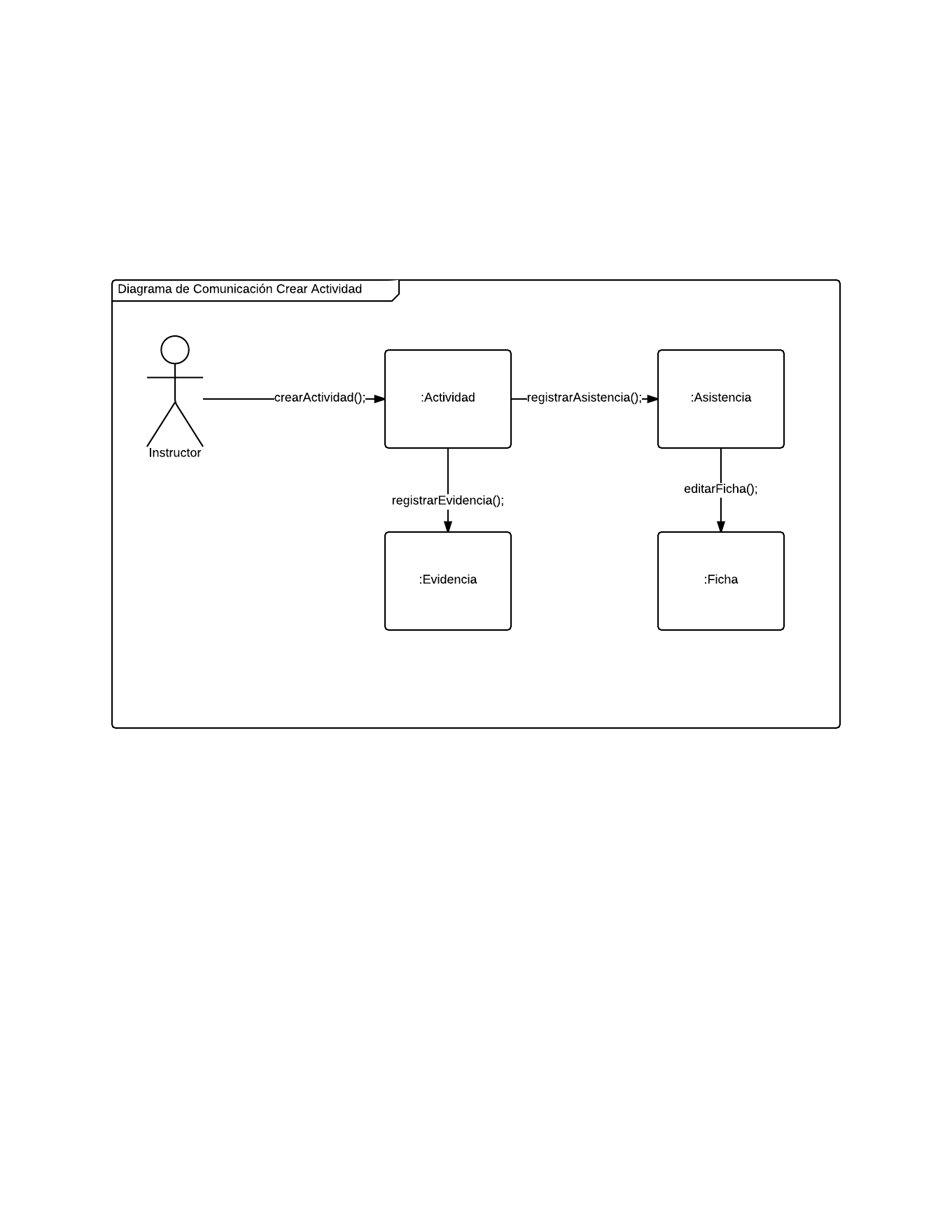






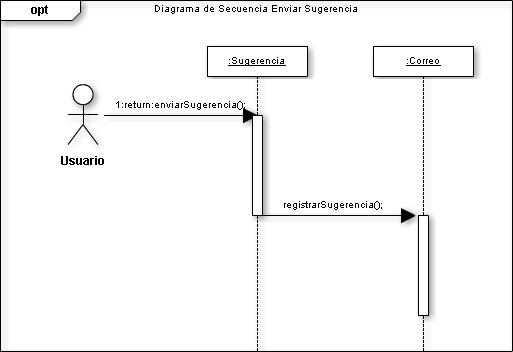


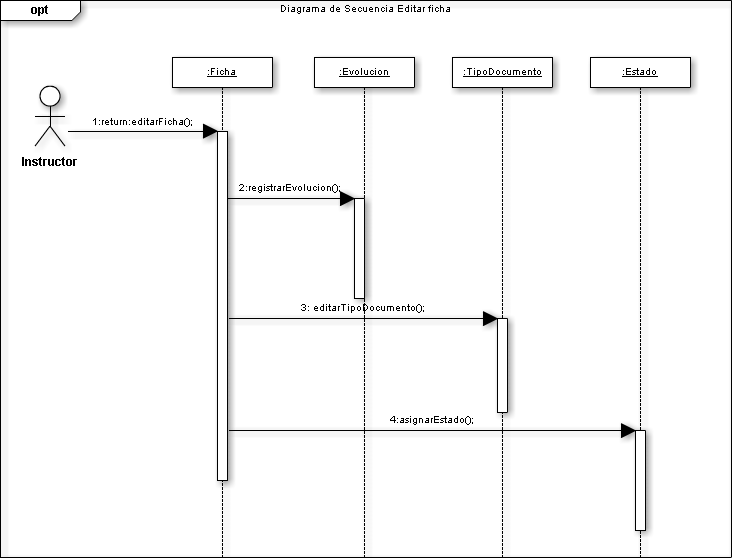


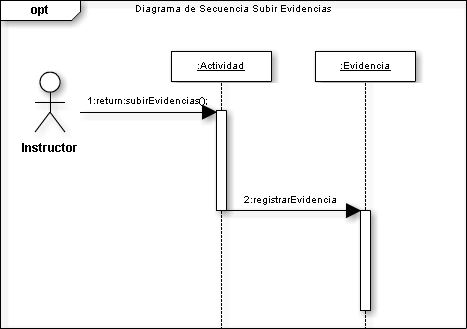


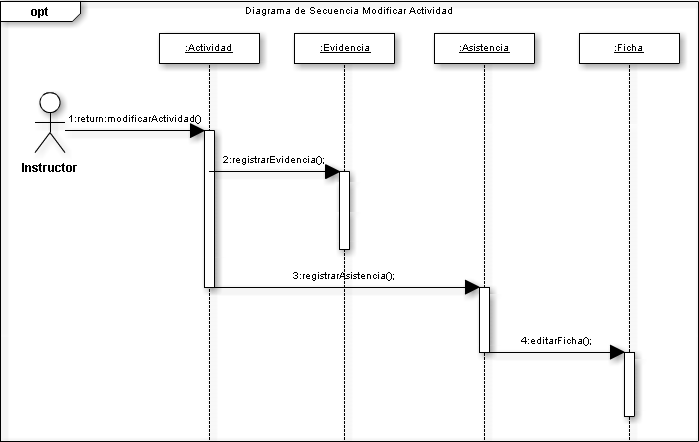
### Diagramas de secuencia

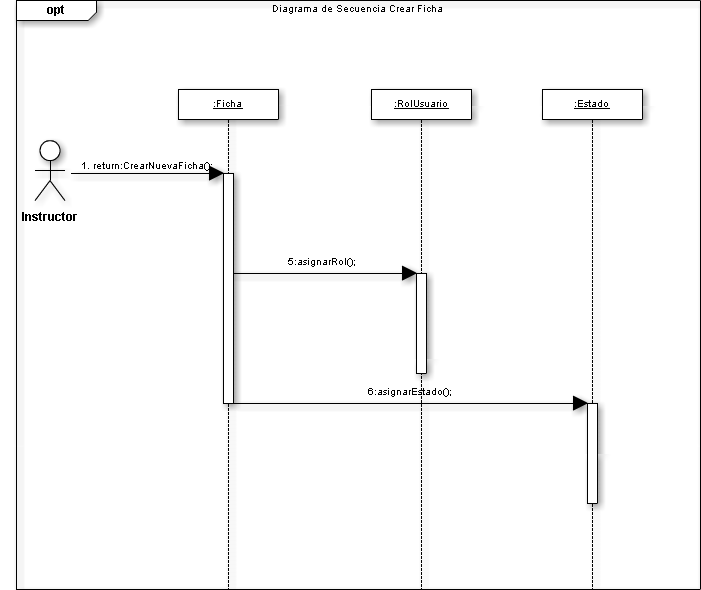
### 





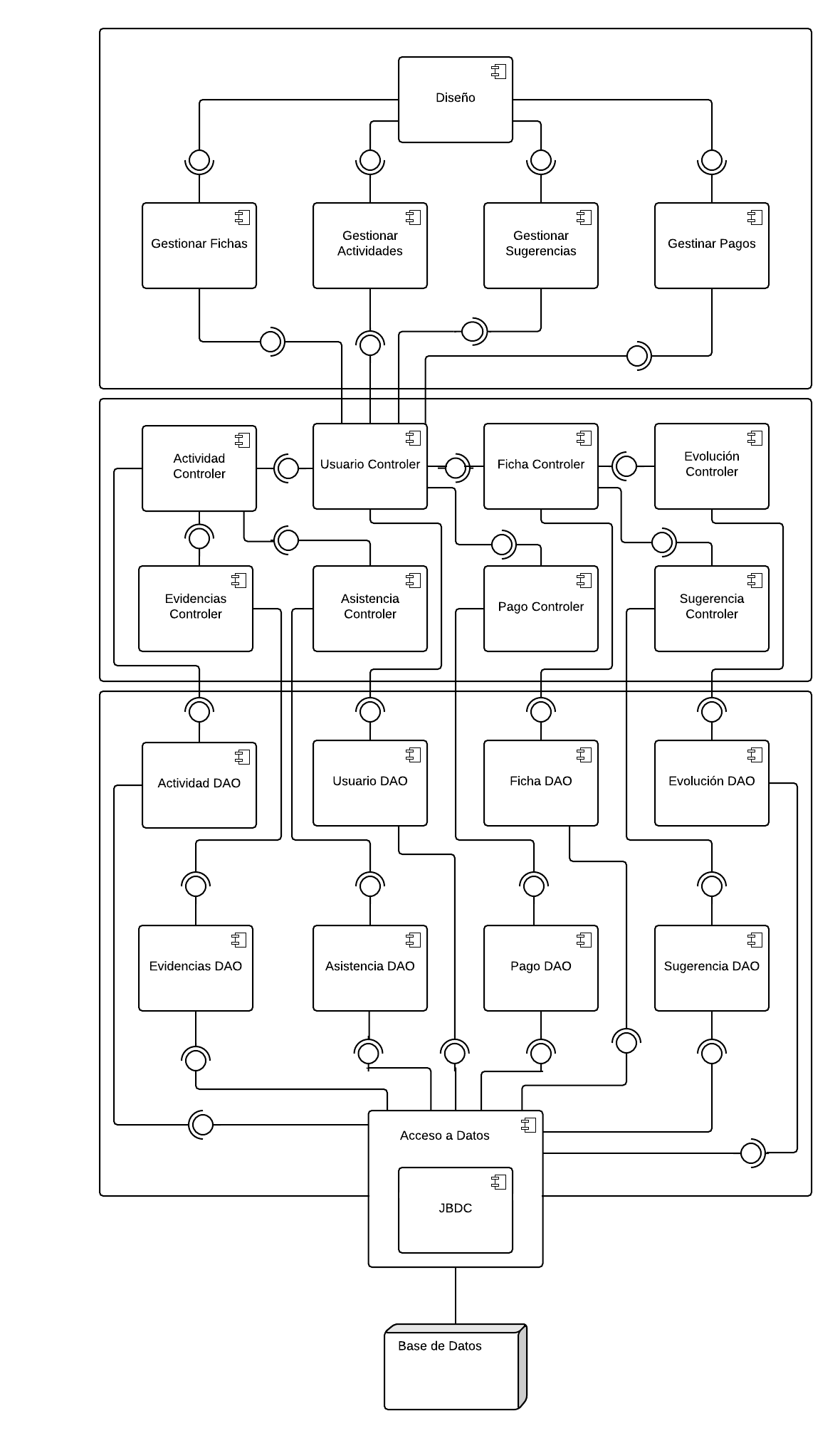




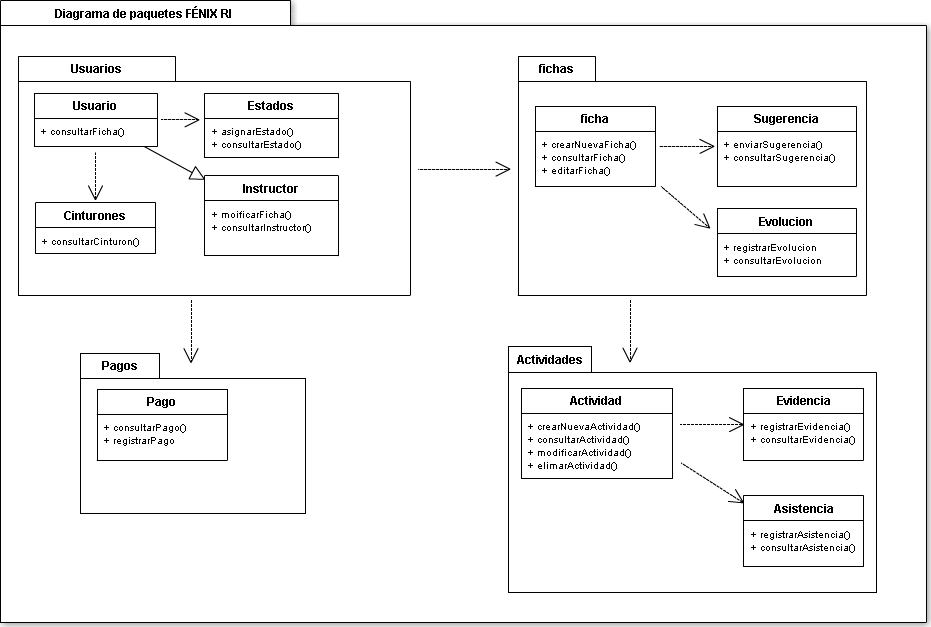


## Vista de Implementación

### Diagrama de Componentes

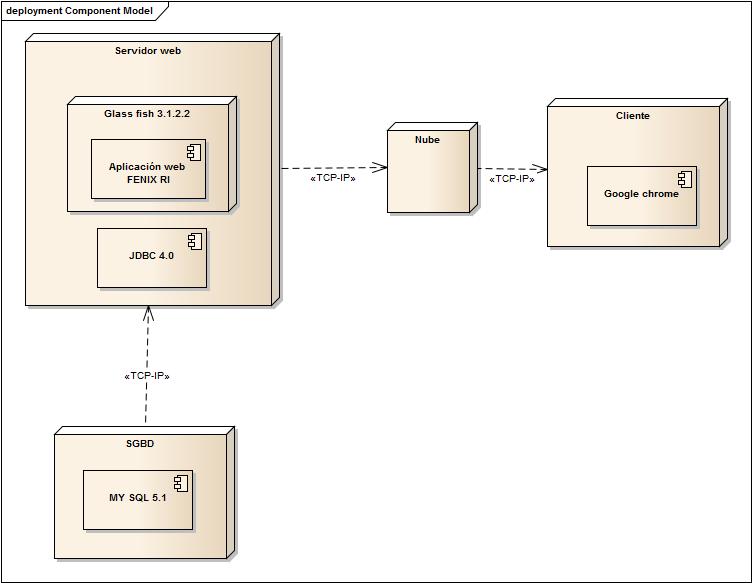


### Diagrama de Paquetes

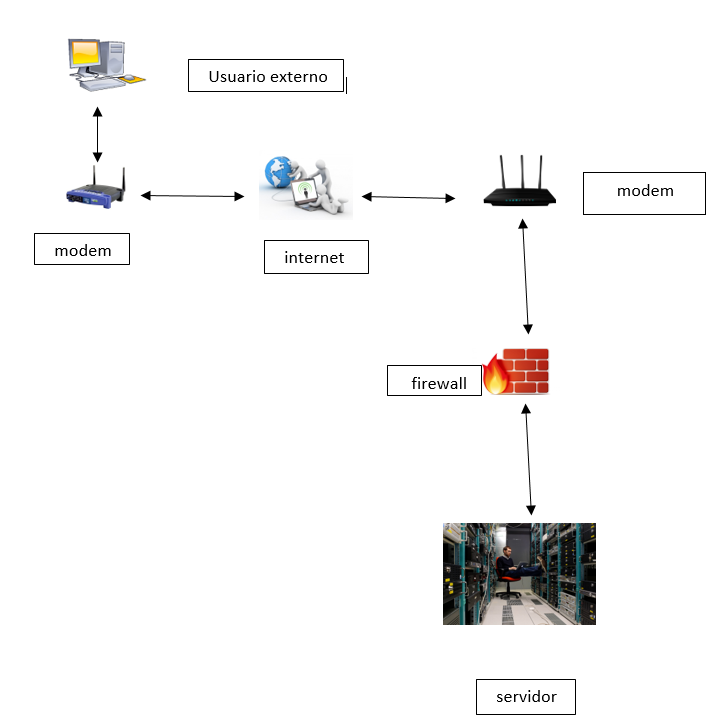


## Vista de Despliegue

### Diagrama de despliegue



### Infraestructura de red



# Arquitectura en capas

**Según las capas de arquitectura:**

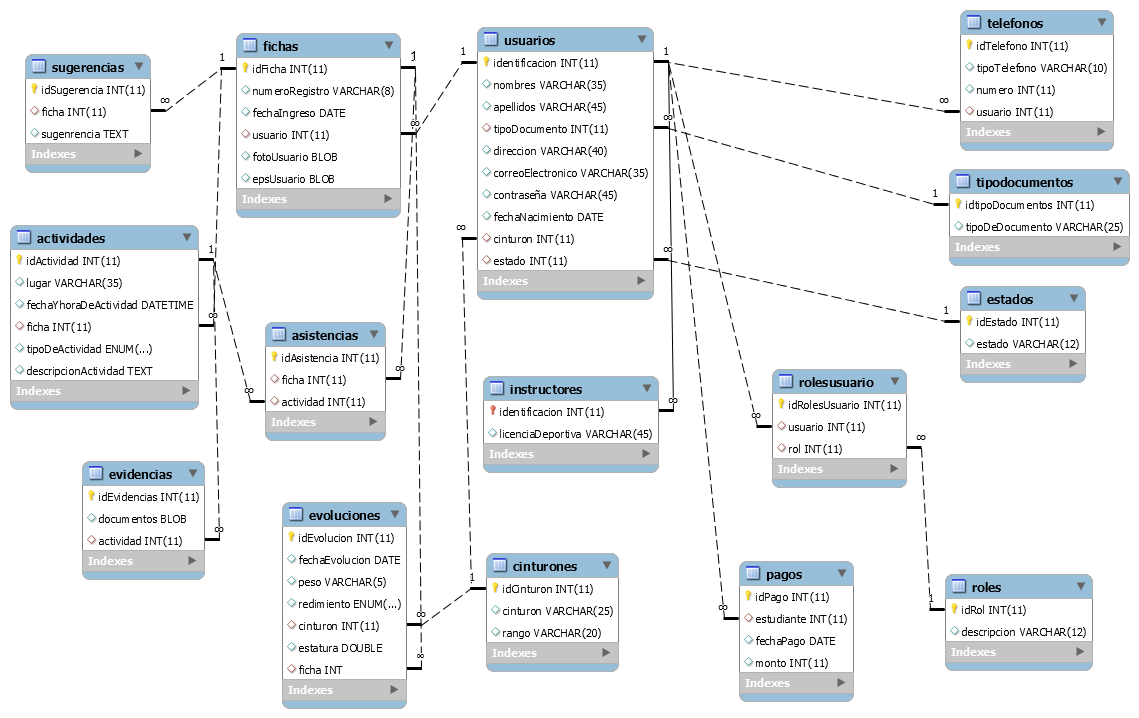
|  |  |
| --- | --- |
| Capas | Componentes |
| Capa de presentación | * HTML5 * Css 3 * Bootstrap v 3.3.7 * JavaScript * JQuery v 3.1.0 |
| Capa de negocio | * Common Annotations * CDI 1.2 * Managed Beans 1.0 * JavaBeans/POJO * Elementos y componentes Enterprise * JavaBeans 3.2 (EJB) |
| Capa de datos | * MySQL workbench 5.1 * Java Data Base connectivity (JDBC) 4.0 * Java transation API (JTA) |

**Según los patrones de desarrollo de software**

|  |  |
| --- | --- |
| **Capas** | **Componentes** |
| Capa de presentación | * DTO Observer– Front Controller |
| Capa lógica de negocio | * DTO, Façade * Business Delegator * Service Locator |
| Capa de persistencia | * DTO Singleton, DAO * Factory Method * Abstract Factory |

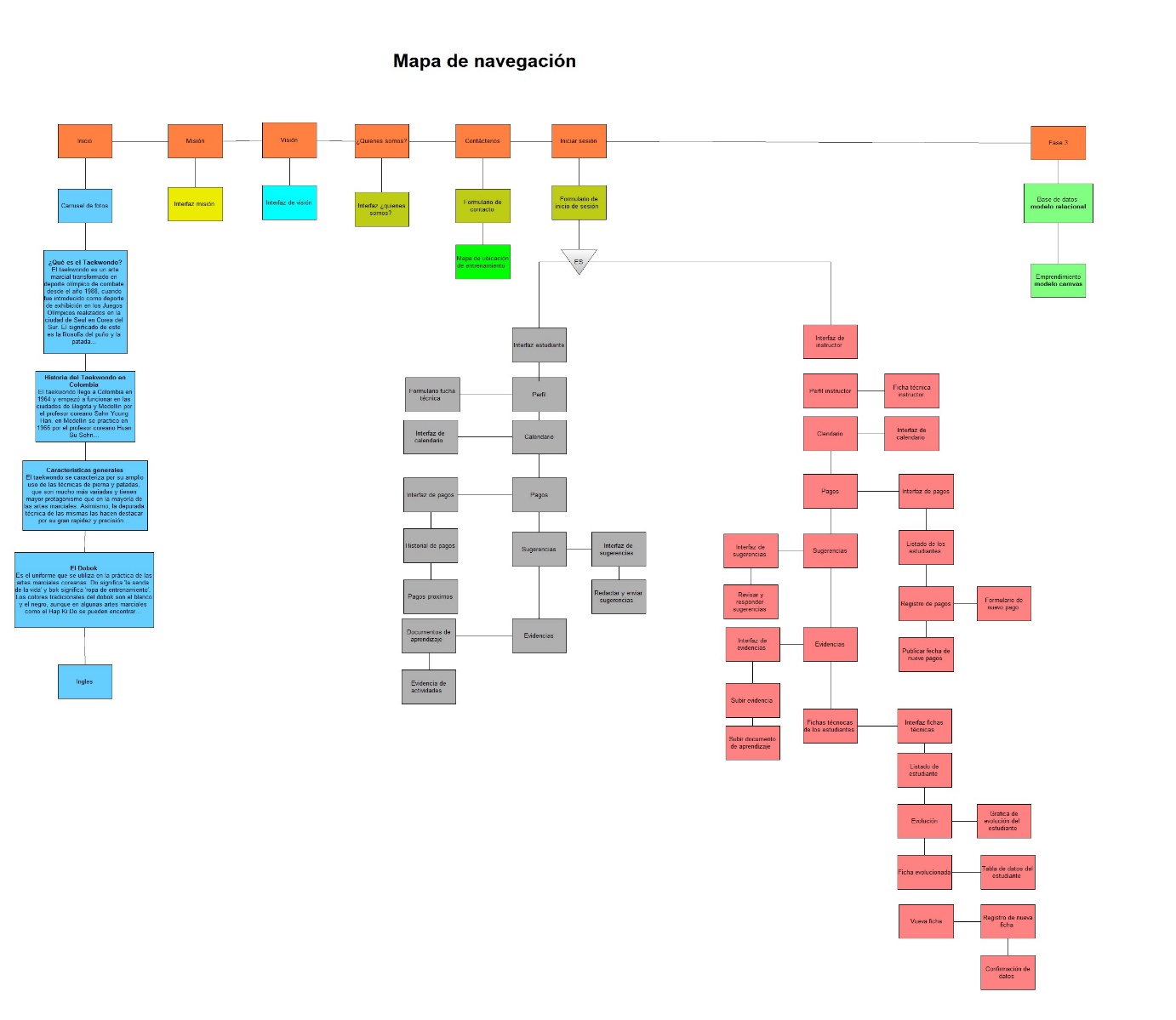
# Vista de Datos

## Modelo Relacional



# Definición de Interfaces de Usuario

Mapa de navegación. Demostración de las interfaces

****

# Características Generales de Calidad

## Tamaño y performance

* El sistema de información permitirá el acceso concurrente de varios usuarios, de modo que puedan realizar simultáneas tareas.
* El tiempo de respuesta en el acceso a la Base de Datos pretende ser de 0.57 segundos para una mejor eficiencia y respuestas a las consultas del cliente
* El tiempo de respuesta de transacciones y validación de los datos de los estudiantes se espera que sea de 0.24 segundos y así ser un sistema con un mejor rendimiento para el almacenamiento de datos.
* Espacio en disco para el cliente será de 350gb
* Espacio en disco para el servidor de Base de datos pretende ser de 350gb

## Calidad

Los aspectos de calidad son importantes en el entorno de la entrega de un proyecto, por esta razón se implementará el sistema de equipo SQA y se pretende tener una revisión y asesoría permanente para la satisfacción del cliente a nivel del producto final.

Para esto nos apegaremos a la reunión de revisión:

1. Revisar al producto, no al programador
2. Se fijará una agenda y se tratará de mantener
3. Se limitará debates entre el grupo y se tendrá en cuenta la opinión de todos para el mejoramiento continuo.
4. Tomar notas sobre las sugerencias del cliente y del equipo SQA.

## Usabilidad

## El sistema de información tendrá una interfaz sencilla, pensada en el usuario y que pueda hacer diferentes tareas sin necesidad de adaptarse al software, además de esto y para su mayor comprensión tendrá alertas al momento de la diligencia de los formularios.

## Eficiencia

## El sistema de información no solo guardará los datos de los estudiantes, sino que también calculará su evolución mediante graficas que medirán su rendimiento cada vez que cambie de cinturón.

Lo que se demostrará al diseñar el sistema de información FÉNIX RI es que la buena planificación y el buen diseño de las interfaces, llevará a un aprovechamiento eficiente de los recursos para obtener un software de calidad.

## Seguridad

## La seguridad para nuestro sistema de información es muy importante y los datos de los clientes lo son también, por eso para proteger sus datos nos apegaremos a la ley orgánica de protección de datos y la ley estatutaria 1581 del 2012 del 17 de octubre, por la cual se dictan disposiciones generales para la protección de datos personales.

## Confiabilidad

## Para una mayor satisfacción de nuestros clientes los datos y seguimientos de estos estará asegurados por la ley constitucional de protección de datos y el manejo de estos será administrada por el instructor que maneja a los estudiantes de taekwondo, solo personas calificadas y propias de club podrán hacerlo.

## Mantenimiento

## El software lleva el procedimiento SQA el cual se enfoca en monitorear el producto en su proceso de desarrollo y corregir errores antes de que se entregue, además de esta manera será más fácil su mantenimiento y seguimiento cuando este se entregue.

## Estándares

Los estándares son muy importantes al momento de desarrollar una aplicación web y en este caso no es la excepción, ya que son la base para un buen producto. Basados en la ISO 9000 e ISO 9001 estándares de calidad de software hemos trabajado y atribuimos nuestro buen manejo del modelo Scrum el cual permite una amplia comunicación entre el equipo de trabajo y la comunicación del cliente, y así poder entregar un producto con los más altos estándares de calidad y eficiencia de respuesta.