<<NOMBRE DEL CLIENTE>>

**Documento de Especificación de Arquitectura**

Realizado por:

William humberto Espinel Corredor

Nicolas Alejandro Cifuentes

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

**Contenido**

[1.](#_2et92p0) Documento de Arquitectura de Software 4

[1.1.](#_tyjcwt) Introducción 4

[1.2.](#_3dy6vkm) Propósito 4

[1.4.](#_4d34og8) Referencias 4

[1.5.](#_2s8eyo1) Definiciones acrónimos y abreviaciones 4

[2.](#_17dp8vu) Generalidades del Proyecto 5

[2.1.](#_3rdcrjn) Problema a Resolver 5

[2.2.](#_26in1rg) Descripción General del Sistema a Desarrollar 5

[2.3.](#_lnxbz9) Identificación de los Stakeholders y sus responsabilidades 5

[3.](#_35nkun2) Vistas de la arquitectura 5

[3.1.](#_1ksv4uv) Vista de Casos de Uso 5

[3.2.](#_44sinio) VISTA DE PROCESOS 5

[3.3.](#_2jxsxqh) VISTA LÓGICA 5

[3.4.](#_3j2qqm3) VISTA DE IMPLEMENTACIÓN 6

[3.5.](#_1y810tw) VISTA DE DESPLIEGUE 6

[4.](#_4i7ojhp) Arquitectura en capas 6

[5.](#_2xcytpi) VISTA DE DATOS 6

[5.1.](#_1ci93xb) Modelo Relacional 6

[6.](#_3whwml4) Definición de Interfaces de Usuario 6

[7.](#_2bn6wsx) Características Generales de Calidad 6

[7.1.](#_qsh70q) Tamaño y performance 6

[7.2.](#_49x2ik5) Calidad 6

[7.3.](#_2p2csry) Usabilidad 7

[7.4.](#_147n2zr) Eficiencia 7

[7.5.](#_3o7alnk) Seguridad 7

[7.6.](#_23ckvvd) Confiabilidad 7

[7.7.](#_ihv636) Mantenimiento 7

[7.8.](#_32hioqz) Estándares 7

1. **Documento de Arquitectura de Software**
   1. **Introducción**
   2. **Propósito**

El sistema de información **SABI** está diseñado para optimizar la interacción entre usuarios interesados en mejorar su salud y bienestar, y profesionales del entrenamiento. SABI facilita la creación y gestión de planes personalizados, mejora la comunicación entre profesionales y clientes, y sistematiza el seguimiento del progreso de los usuarios.

Al automatizar la creación de planes, se reduce significativamente el tiempo y esfuerzo que los profesionales necesitan para desarrollar programas de entrenamiento, lo que les permite ofrecer un servicio más ágil y eficiente. Además, se proporciona un canal centrado en la comunicación entre clientes y profesionales, mejorando la disponibilidad y rapidez en la respuesta, y eliminando la necesidad de reuniones presenciales o dependencias de múltiples medios de contacto.

El seguimiento del progreso del usuario se simplifica permitiendo un registro digital y organizado de los avances en rutinas. Esto no solo facilita a los usuarios el monitoreo de su progreso, sino que también permite a los profesionales acceder a esta información de manera rápida y sencilla, lo que mejora la precisión y efectividad de los ajustes que realicen en los planes personalizados.

* 1. **Referencias**

<https://gymvirtual.com>

<https://www.entrenamientos.com>

<https://significadosweb.com/definicion-de-rutina-que-es-ejemplos-tipos-y-para-que-sirve-sinonimo-y-significado/>

<https://ejemplosverdes.com/ejemplos-de-rutina-definicion-segun-autor-que-es-concepto-significado/>

<https://definicion.edu.lat/definicion/rutina.html>

<https://www.um.es/web/medicinadeportiva/contenido/planificacion/pruebas#:~:text=Son%20las%20pruebas%20que%20realizamos,del%20estado%20actual%20del%20deportista>.

<https://es.wikipedia.org/wiki/Entrenador>

<https://definicion.de/entrenador/>

* 1. **Definiciones acrónimos y abreviaciones**

**ARQUITECTURA DE SOFTWARE:** conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCIÓN DE ARQUITECTURA**: colección de productos de documentación.

**VISTAS:** es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

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

1. **Generalidades del Proyecto**
   1. **Problema a Resolver**

Actualmente, los procesos relacionados con la creación de planes personalizados, el contacto entre clientes y profesionales del entrenamiento y el seguimiento del progreso de los usuarios son ineficientes y desorganizados. Estos problemas afectan la calidad del servicio que los profesionales pueden ofrecer y dificultan el compromiso y el progreso de los usuarios en sus objetivos de salud y bienestar.

* 1. **Descripción General del Sistema a Desarrollar (General y por módulo)**

Desarrollar un software para gestionar y controlar los hábitos de vida de los usuarios

* Gestionar el diagnóstico inicial del estado de los clientes
* Facilitar el seguimiento de la realización de los planes del cliente
* Generar un reporte de resultados por usuario
  1. **Identificación de los Stakeholders y sus responsabilidades**

| **STAKEHOLDER** | **DESCRIPCIÓN** |
| --- | --- |
| Integrante de equipo (Gerente del proyecto) | Responsable de la planificación, ejecución y cierre del proyecto. coordina las actividades del equipo y asegura que se cumplan según los plazos y objetivos |
| Integrante de equipo (Desarrollador) | Encargado de la implementación técnica del proyecto, escribe el código y trabaja en el desarrollo de la aplicación o sistema |
| Integrante de equipo (Diseñador UX/UI) | Se ocupa de la experiencia del usuario del diseño de la interfaz. Asegurando que el producto sea intuitivo y visualmente atractivo |
| Integrante de equipo (Analista de negocios) | Define los requisitos del proyecto asegurando que el producto final cumpla con las expectativas |
| Integrante de equipo (Tester) | Responsable de probar el producto para identificar errores y asegurar que cumpla con los estándares de calidad antes de su lanzamiento |

1. **Vistas de la arquitectura**
   1. **Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el software)**

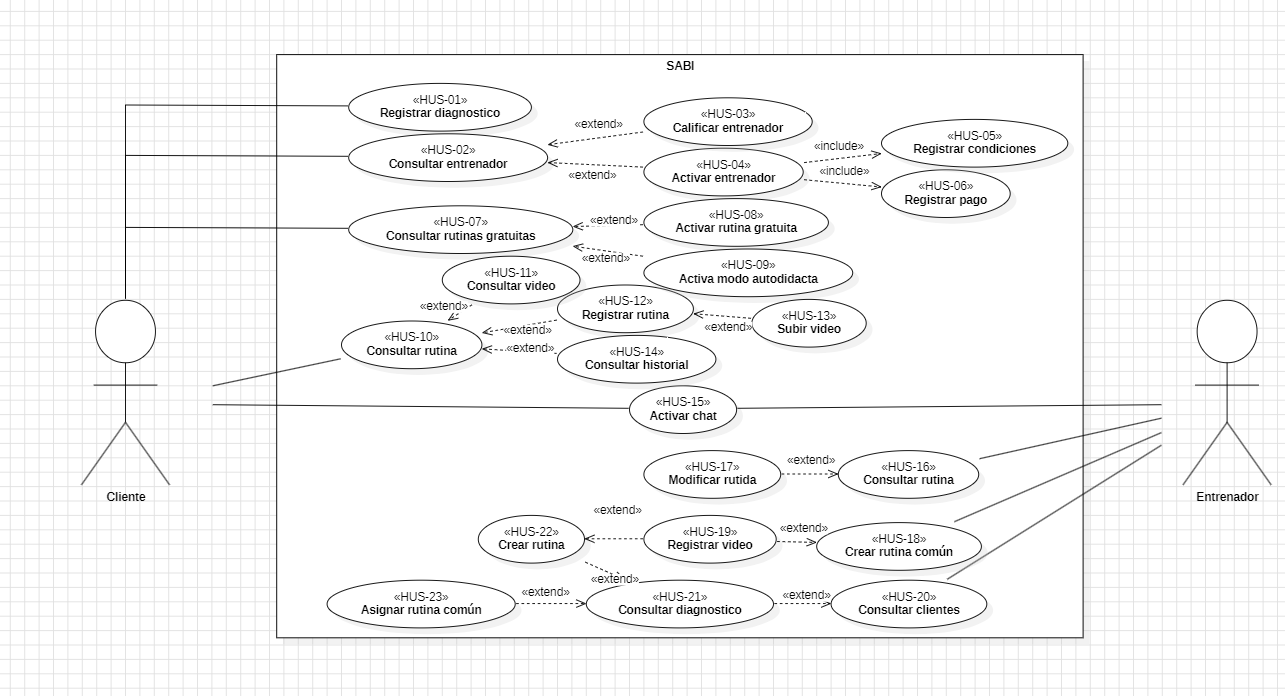


Figura 1 [Diagrama Casos de Uso](https://drive.google.com/file/d/1lg_USSKYflLyI1lWdJZQveGdPMYltJUE/view?usp=sharing)

**Diagnóstico y Registro**: El cliente comienza por **registrar un diagnóstico** (HUS-01), donde ingresa sus medidas y condiciones físicas. Este diagnóstico es consultado por el entrenador, quien lo usa para ajustar o **crear rutinas** personalizadas (HUS-18).

**Consulta de Entrenadores**: Los clientes pueden **consultar la información de los entrenadores** (HUS-02) disponibles en la plataforma, incluyendo su especialidad, certificaciones y calificaciones, lo que les permite tomar una decisión informada antes de suscribirse.

**Acceso a Rutinas**: Los clientes tienen la opción de **consultar rutinas gratuitas** (HUS-07), pero también pueden acceder a las rutinas **personalizadas** (HUS-10) que han sido creadas por el entrenador con base en el diagnóstico y objetivos personales. Dentro de la consulta de rutina, los clientes pueden visualizar los **videos instructivos** (HUS-11), asegurando que entiendan cómo realizar correctamente cada ejercicio.

**Modificación y Actualización de Rutinas**: Los entrenadores pueden **modificar rutinas** (HUS-17) en cualquier momento si consideran que el cliente necesita ajustes en su programa. Además, pueden crear rutinas genéricas o **comunes** (HUS-23) que se asignen a varios clientes que compartan características o metas similares.

**Historial y Progreso**: Los clientes también pueden revisar su **historial de rutinas** y diagnósticos previos (HUS-14), lo que les permite monitorear su progreso a lo largo del tiempo.

**Videos y Comunicación**: Los entrenadores pueden **subir videos** (HUS-13) que los clientes utilizan para guiarse durante los ejercicios. Además, ambos actores pueden **activar un chat** (HUS-15) para comunicarse directamente, lo que facilita la resolución de dudas o la motivación en tiempo real.

**Pagos y Condiciones**: A nivel administrativo, los entrenadores pueden **registrar pagos** (HUS-06) realizados por los clientes y también registrar **condiciones físicas o médicas** (HUS-05) para ajustar las rutinas de manera precisa.

* 1. **VISTA DE PROCESOS**

Todos los diagramas que hayan elaborado en **fase II, p**or módulo (aplica para los puntos 3.2.1 al 3.3.2), debidamente argumentados cada uno de los diagramas.

* + 1. **Diagrama de Actividades**

3.2.1.1. Modulo Diagnostico

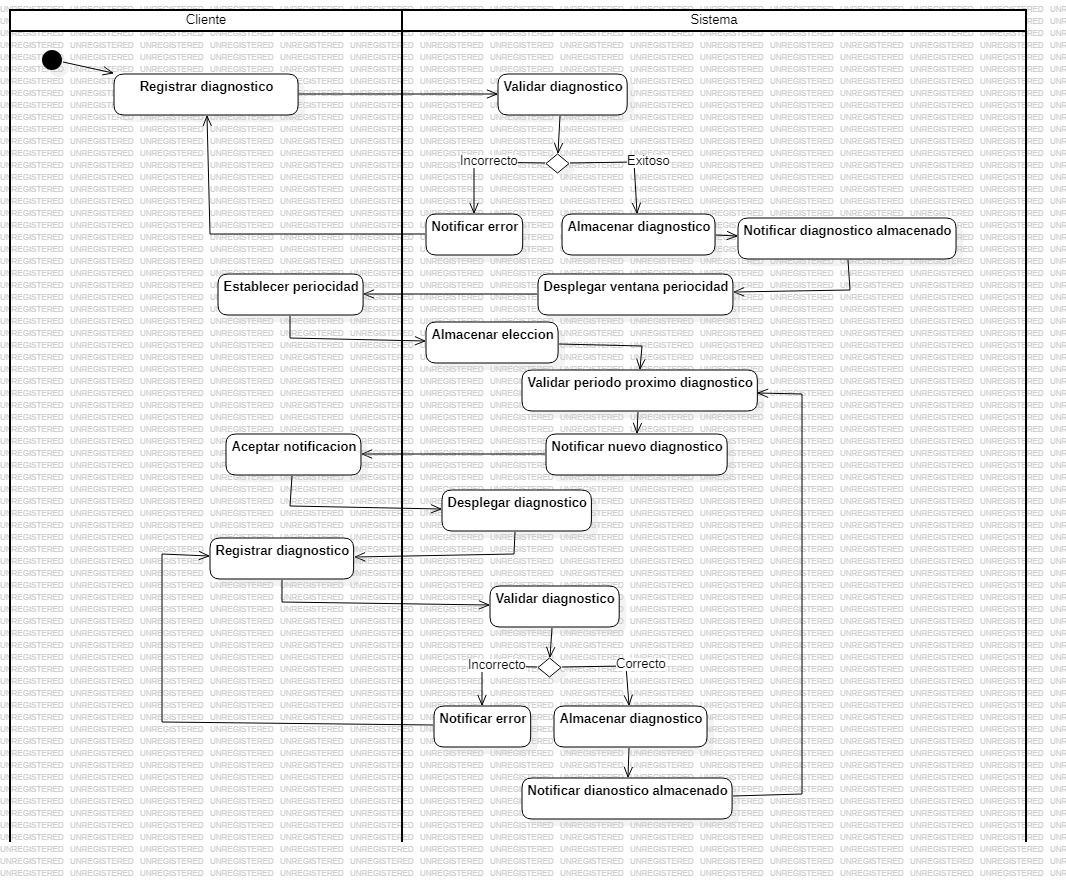


Figura 2 [Diagrama de Actividades Diagnóstico](https://drive.google.com/file/d/1ypm7OTVn554yfPBLYhdal9gkwuqqpfqH/view?usp=sharing)

Este diagrama de actividades muestra cómo un cliente registra sus diagnósticos en el sistema y cómo este gestiona la validación, almacenamiento y notificación de los diagnósticos. También permite al cliente establecer la periodicidad para próximos diagnósticos, lo que crea un ciclo continuo de registro y validación. Cada vez que el sistema valida y almacena correctamente un diagnóstico, notifica al cliente y prepara la información para futuros diagnósticos, lo que facilita un seguimiento regular de la salud o el estado físico del cliente.

3.2.1.2.1. Módulo Seguimiento 1

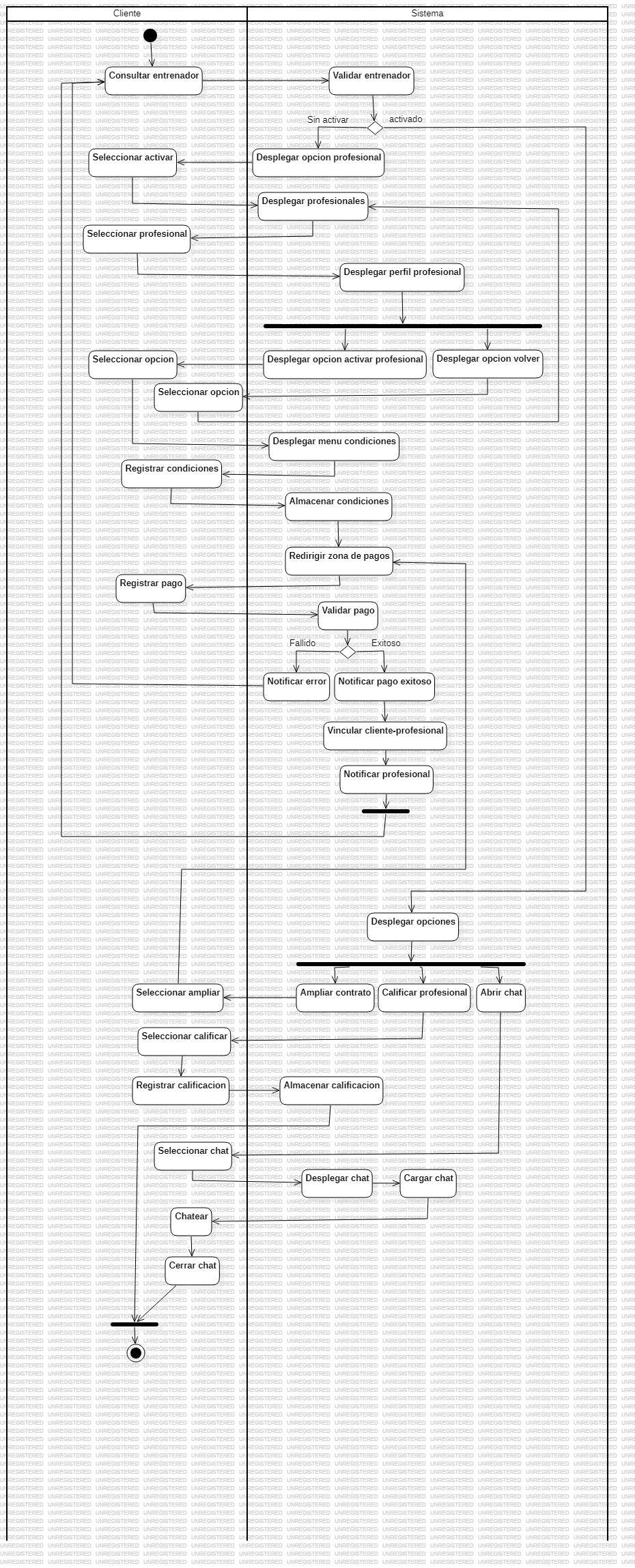


Figura 3 [Diagrama de Actividades Consultar profesional](https://drive.google.com/file/d/1iut3Cp7kMlvxOlz1eHYwPcWf0atFqkUS/view?usp=sharing)

Este proceso muestra la interacción entre un **cliente** y el **sistema** para consultar y seleccionar a un entrenador profesional. Los pasos clave son los siguientes:

1. **Consultar Entrenador**: El cliente comienza solicitando consultar a un entrenador.
2. **Validar Entrenador**: El sistema valida si el entrenador está activado o no.
   * Si está activado, despliega la opción de consultar profesionales.
   * Si no está activado, la consulta no avanza.
3. **Seleccionar Profesional**: El cliente selecciona un profesional.
4. **Desplegar Perfil del Profesional**: El sistema muestra el perfil del profesional.
5. **Activar o Volver**: El sistema ofrece opciones para activar al profesional o volver al menú anterior.
6. **Registrar Condiciones y Pago**: Si el cliente decide seguir, selecciona condiciones y paga, lo cual el sistema valida:
   * Si el pago es exitoso, el sistema vincula al cliente con el profesional y notifica a ambas partes.
   * Si el pago falla, se notifica un error.
7. **Ampliar Contrato, Calificar o Chatear**: Una vez activado el servicio, el cliente puede elegir:
   * Ampliar el contrato con el profesional.
   * Calificar al profesional.
   * Abrir un chat para comunicación directa.

3.2.1.2.2. Módulo Seguimiento 2

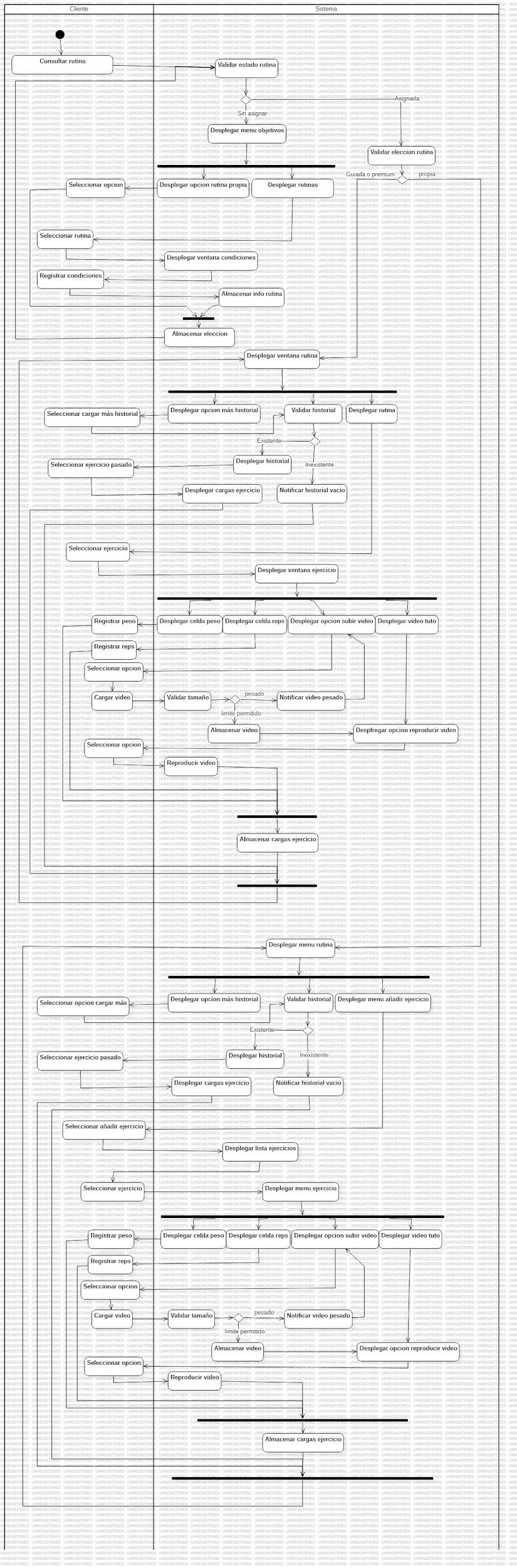


Figura 4 [Diagrama de Actividades Seguimiento](https://drive.google.com/file/d/1Qj8YvZRcgZIE5-dWPI03GC4ji0iiNNPf/view?usp=sharing)

Este diagrama ilustra el seguimiento de una rutina de ejercicios por parte de un cliente, apoyado por el sistema. Los pasos clave son los siguientes:

1. **Consultar Rutina**: El cliente inicia solicitando una rutina de ejercicios.
2. **Validar Estado de Rutina**: El sistema valida si la rutina ya ha sido asignada:
   1. Si no está asignada, el sistema despliega un menú de objetivos para que el cliente elija.
   2. Si está asignada, el sistema despliega la rutina.
3. **Seleccionar Rutina y Registrar Condiciones**: El cliente elige una rutina, registra las condiciones y el sistema almacena esa información.
4. **Seguimiento del Ejercicio**:
   1. El cliente puede seleccionar un ejercicio y registrar los pesos y repeticiones.
   2. También puede cargar un video de su ejercicio y el sistema valida el tamaño del archivo antes de almacenarlo.
   3. Si el video es demasiado grande, el sistema notifica un error, de lo contrario, permite la reproducción.
5. **Historial de Ejercicios**: El cliente puede consultar el historial de ejercicios y revisar sus registros previos. Si no existen registros, el sistema notifica que el historial está vacío.

3.2.1.3. Modulo Control

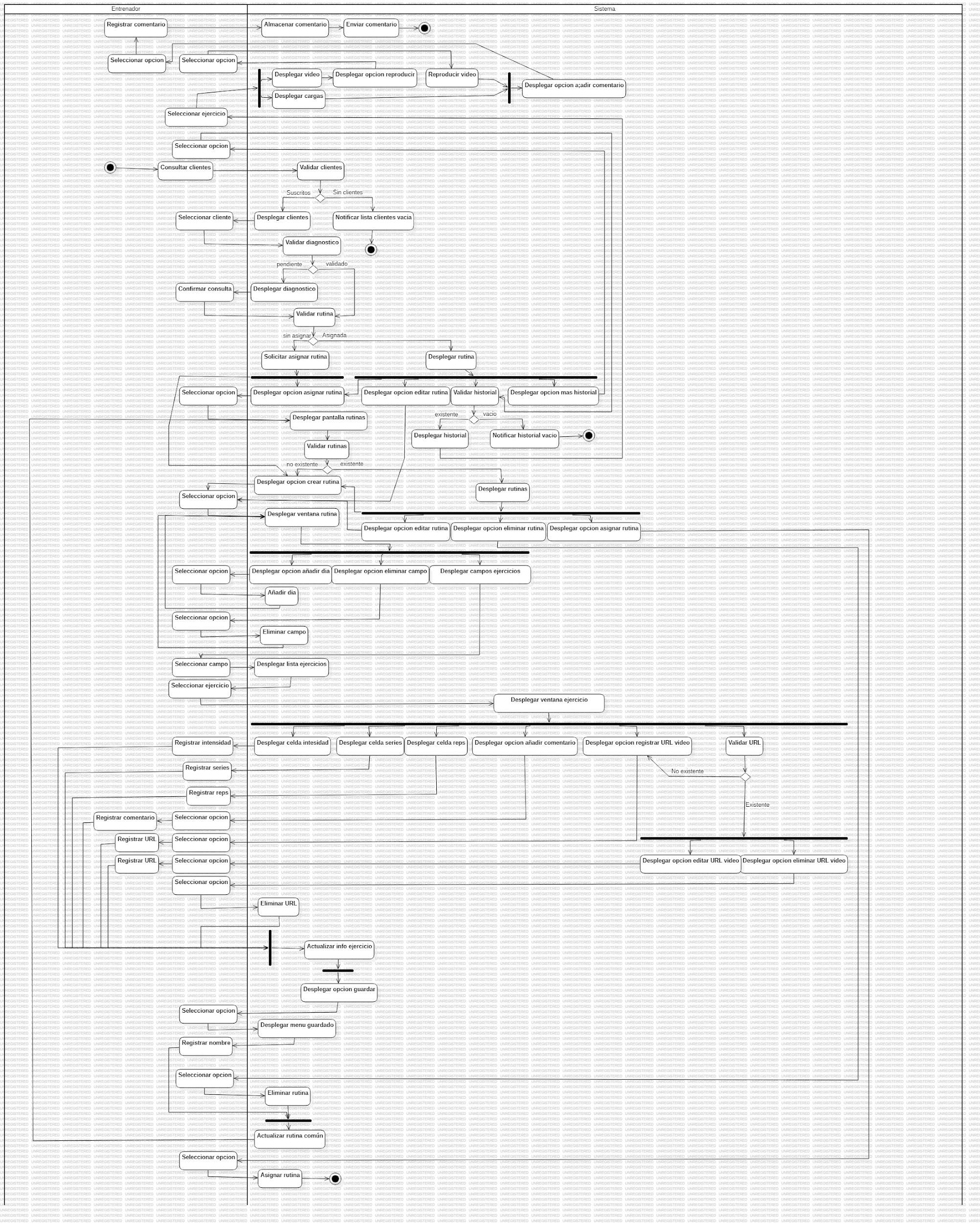


Figura 5 [Diagrama de Actividades Control](https://drive.google.com/file/d/1-5C0S_3sOxmtVLjhihUVFjixhzB45DZL/view?usp=sharing)

Este diagrama de actividades representa el flujo que sigue un entrenador al interactuar con una página web para gestionar sus clientes y las rutinas que les asigna. A lo largo del diagrama se identifican las acciones principales y las posibles decisiones que puede tomar el entrenador, lo que permite una interacción dinámica y flexible. A continuación, te detallo el proceso de forma más específica:

1. Ingreso a la plataforma: El entrenador accede al sistema, donde tiene la opción de gestionar sus clientes y las rutinas que ellos realizan.
2. Selección de cliente: Una de las primeras acciones es la selección de un cliente específico desde una lista. Esta lista puede estar compuesta por clientes que ya existen en el sistema o puede optar por crear un nuevo cliente.
3. Consulta de información del cliente: Luego de seleccionar al cliente, el entrenador puede ver detalles adicionales del mismo, como sus rutinas previas o historial de actividades.
4. Creación y modificación de rutinas: Una vez que un cliente es seleccionado, el entrenador tiene la opción de crear nuevas rutinas personalizadas. Para ello, puede seleccionar entre diferentes ejercicios, agregar series, repeticiones, y otros parámetros que considere importantes.
5. Opciones multimedia: El entrenador puede agregar contenido multimedia, como videos de ejercicios, para guiar mejor al cliente en su rutina. También tiene la posibilidad de reproducir estos videos y añadir comentarios o audios explicativos.
6. Validación y edición: En cualquier momento, el entrenador puede validar que los datos sean correctos, realizar ediciones o incluso eliminar rutinas si lo considera necesario.
7. Registro de historial: Existe también una función para registrar un historial de rutinas pasadas y validarlas para asegurar que los cambios se guarden correctamente.

Este flujo detalla cómo el entrenador interactúa de forma cíclica con el sistema, ajustando y personalizando las rutinas para cada cliente según las necesidades y características particulares, con opciones de validar, editar, eliminar o reproducir contenido multimedia.

* 1. **VISTA LÓGICA**
     1. **Diagramas - Clases (Actualizado)**

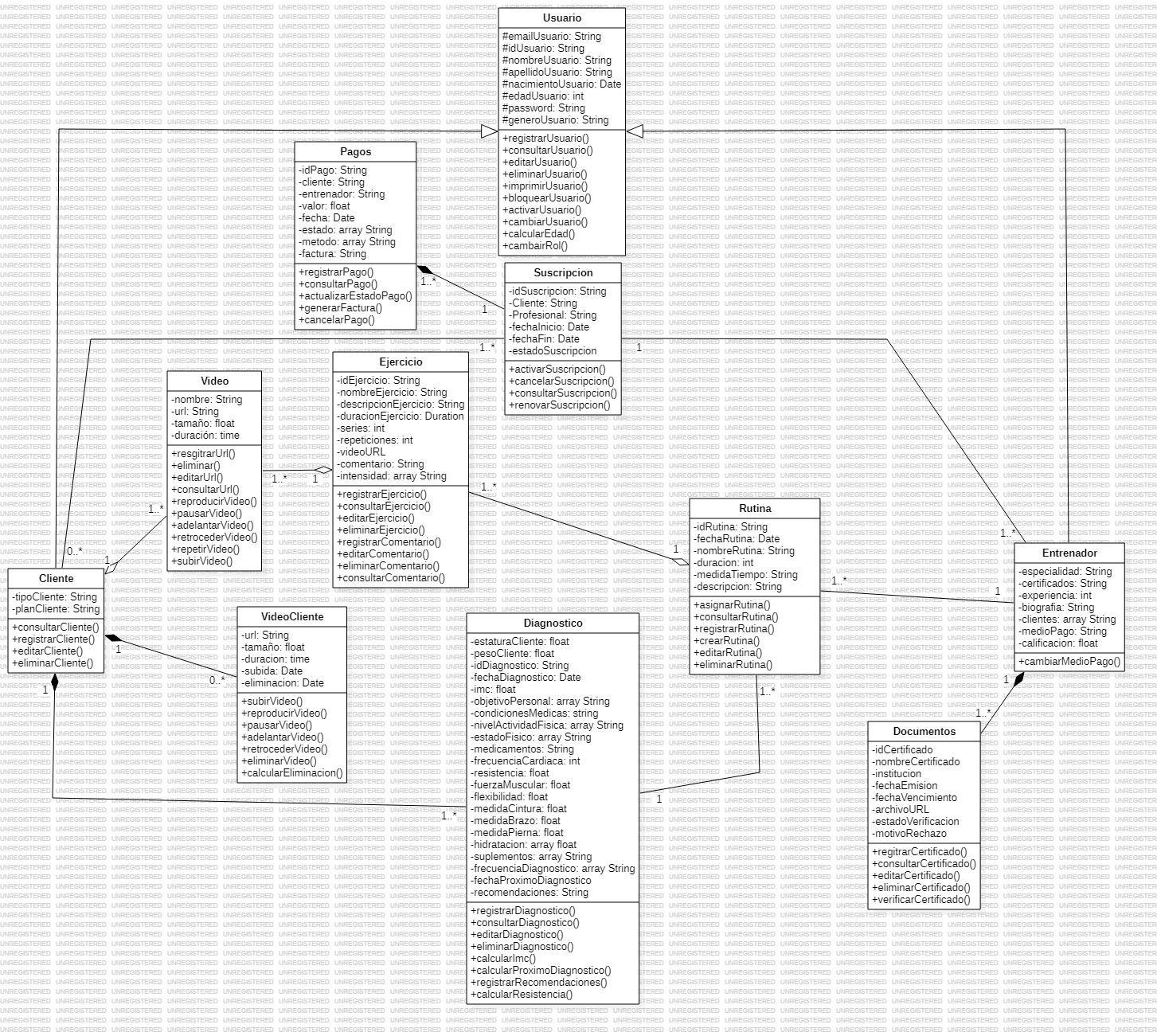


figura 6 [Diagrama de Clases](https://drive.google.com/file/d/1f2jquNwaFr7nDGfHlWsGwOP2S1uQnxp7/view?usp=sharing)

### 1. Gestión de Usuarios y Roles

El sistema permite manejar diferentes tipos de usuarios, principalmente **entrenadores** y **clientes**, que heredan las características de la clase **Usuario**. Cada usuario tiene sus datos personales, los cuales pueden ser registrados, actualizados o eliminados por los métodos dentro de esta clase. Los entrenadores tienen un perfil más especializado con atributos adicionales como certificaciones, biografía, medios de pago, y una calificación en función de sus clientes.

### 2. Entrenadores y sus Clientes

Los entrenadores pueden gestionar a sus **clientes** mediante las siguientes funciones:

* **Consultas de clientes**: Pueden ver la información de sus clientes y sus diagnósticos.
* **Asignación de rutinas**: Pueden crear, editar o eliminar rutinas personalizadas para cada cliente. Cada rutina incluye ejercicios y, posiblemente, contenido multimedia asociado (videos de ejercicios, por ejemplo).

### 3. Rutinas y Ejercicios

Las **rutinas** están compuestas de varios **ejercicios**, que se crean y configuran según los objetivos del cliente. El sistema permite:

* **Crear y asignar rutinas** a un cliente, con una duración definida y un tiempo promedio estimado.
* **Adjuntar videos o instrucciones multimedia** a cada ejercicio dentro de la rutina, lo que ayuda al cliente a comprender la correcta ejecución.
* **Modificar o eliminar rutinas** según el progreso del cliente.

### 4. Diagnósticos Personalizados

Los entrenadores pueden realizar un **diagnóstico** físico del cliente, que incluye datos como estatura, peso, flexibilidad, índice de masa corporal, etc. Esto permite adaptar mejor las rutinas y los ejercicios a las necesidades específicas de cada cliente. El diagnóstico puede actualizarse en el tiempo y el sistema calcula automáticamente ciertos parámetros, como la resistencia o recomendaciones de entrenamiento.

### 5. Suscripciones y Pagos

El sistema maneja un esquema de **suscripciones**, que están vinculadas a los **pagos**. Cada cliente tiene una suscripción activa que puede ser renovada o cancelada. Además, hay un manejo completo de los pagos:

* **Registrar pagos y consultar facturas**.
* **Validar el estado de una suscripción** para determinar si el cliente puede seguir utilizando los servicios del entrenador.

### 6. Multimedia (Videos)

Los entrenadores pueden subir videos explicativos para los ejercicios dentro de una rutina. Estos videos están relacionados tanto a los **ejercicios** como a los **clientes**. El sistema permite:

* **Subir, editar, y eliminar videos**.
* **Reproducir los videos** cuando el cliente lo necesite durante el entrenamiento.

### 7. Documentos y Certificaciones

Los entrenadores también tienen la capacidad de gestionar sus **certificaciones** a través del sistema. Estas certificaciones son documentos que pueden ser subidos, verificados y eliminados. Esto garantiza que los entrenadores estén debidamente calificados y certificados, lo que es crucial para la reputación y confianza en el sistema.

* + 1. **Diagrama - Objetos**

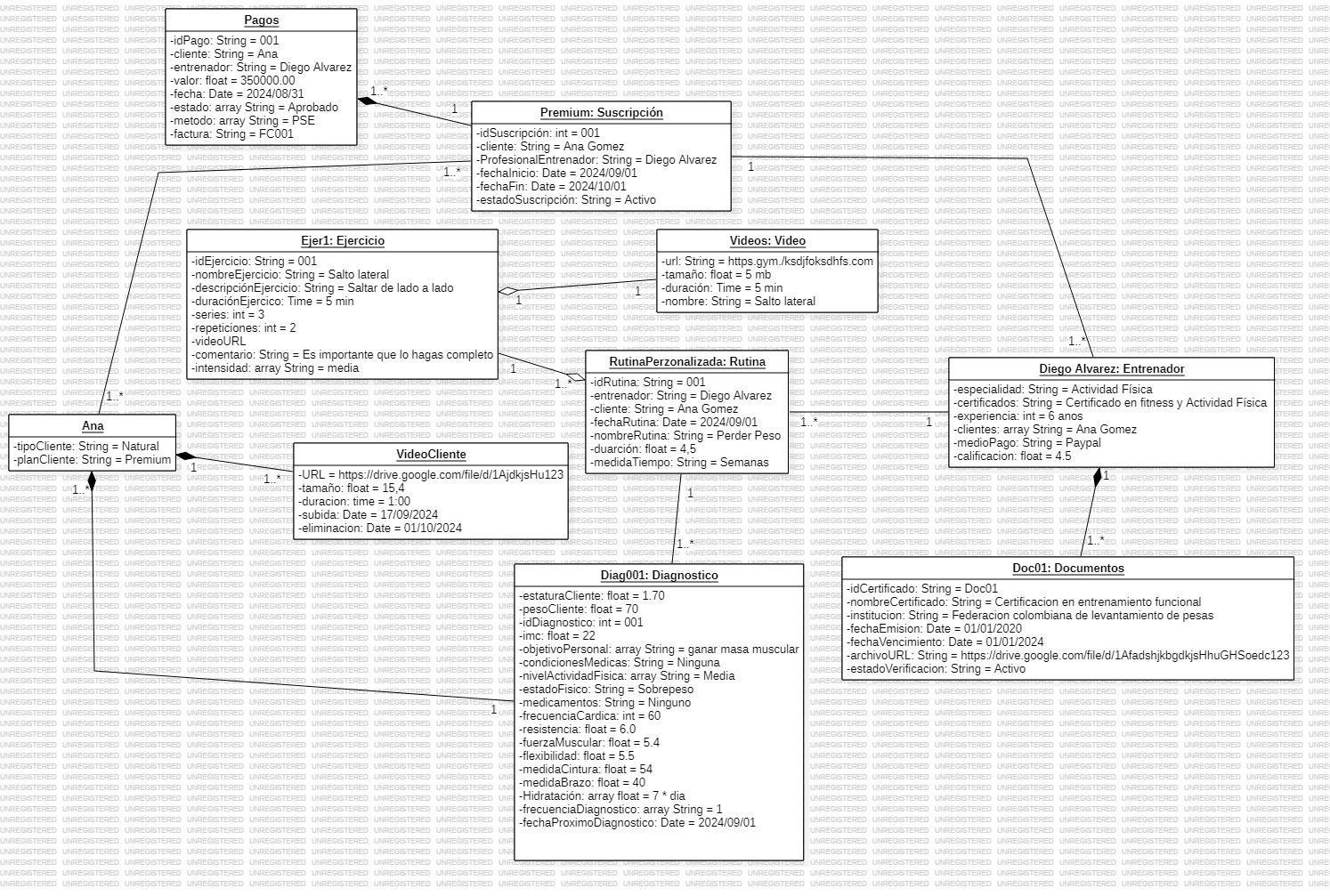


Figura 7 [Diagrama de Objetos](https://drive.google.com/file/d/18fmuw1IcD8GbthaLYD4qLRCoPMdHvjxs/view?usp=sharing)

En el sistema, cada **cliente** (como **Ana Gómez**) tiene un perfil que detalla sus características físicas y de salud, registradas a través de un **diagnóstico** inicial. Este diagnóstico incluye medidas como la estatura, peso, IMC (Índice de Masa Corporal), nivel de actividad física, objetivos personales (como **ganar masa muscular**) y el estado de salud general, información crucial para personalizar sus rutinas de ejercicio.

Los **entrenadores**, como **Diego Álvarez**, están asociados a cada cliente a través de una **suscripción premium** que permite acceso a rutinas y seguimiento personalizado. Los entrenadores tienen certificaciones y experiencia que el cliente puede consultar antes de contratar sus servicios. Cada entrenador maneja un listado de clientes y puede cobrar por sus servicios a través de métodos como **PayPal**. También cuentan con una **calificación** por parte de sus clientes, que refleja su rendimiento y profesionalismo.

El sistema incluye módulos que permiten a los clientes realizar pagos a través de plataformas como **PSE**, y cada pago está vinculado a una **factura** que refleja los servicios recibidos, como el acceso a rutinas personalizadas y suscripciones.

* + 1. **Diagramas - Secuencia**

**15.1.3.1. Seguimiento**

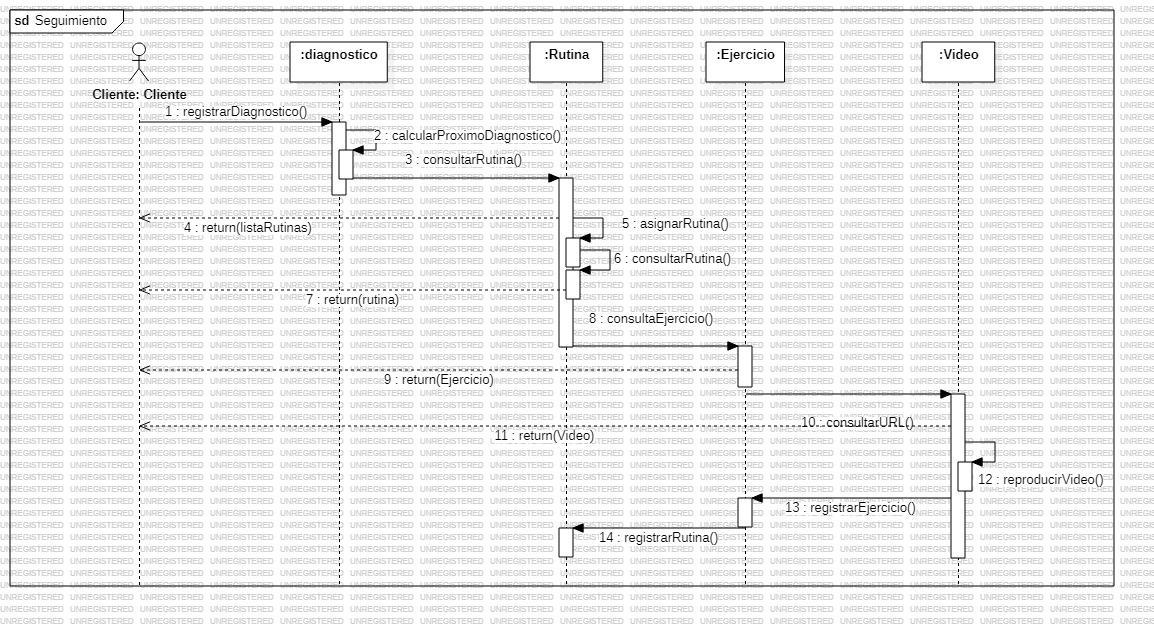


Figura 8 [Diagrama de Secuencias Seguimiento](https://drive.google.com/file/d/1znFf8TXVp7Td0kZ8_JeLEKB-F8KxN7Y4/view?usp=sharing)

Este diagrama de secuencia describe cómo el sistema sigue el flujo de interacción para la **gestión del seguimiento de un cliente**, desde el registro de un diagnóstico hasta la asignación y consulta de rutinas y ejercicios, involucrando videos asociados. A continuación, te explico cómo funciona este flujo paso a paso:

### 1. Registro del Diagnóstico:

* El **cliente** inicia el proceso solicitando el **registro de un diagnóstico** (registrarDiagnostico()).
* La clase **Diagnóstico** calcula el próximo diagnóstico (calcularProximoDiagnostico()) basándose en los datos actuales del cliente.

### 2. Consulta de Rutinas:

* El sistema consulta la rutina existente para el cliente llamando al método consultarRutina() en la clase **Rutina**.
* El método consultarRutina() devuelve una **lista de rutinas** asociadas al cliente (return(listaRutinas)).

### 3. Asignación de Rutinas:

* Si no se encuentra una rutina adecuada o si es necesario, el entrenador puede **asignar una nueva rutina** al cliente (asignarRutina()).
* El sistema verifica las rutinas asignadas a través de la consulta a la clase **Rutina** (consultarRutina()), y finalmente devuelve la rutina asignada o consultada (return(rutina)).

### 4. Consulta y Asignación de Ejercicios:

* Una vez asignada la rutina, el sistema consulta los **ejercicios** que forman parte de la misma (consultarEjercicio()).
* El sistema devuelve los ejercicios correspondientes (return(Ejercicio)).
* En caso de que no se haya asignado previamente un ejercicio, el entrenador puede **asignar un ejercicio** (asignarEjercicio()).

### 5. Consulta de Videos de Ejercicios:

* Algunos ejercicios tienen **videos asociados** que explican cómo realizarlos. El sistema **consulta la URL del video** (consultarURL()), obteniendo el recurso multimedia que acompaña al ejercicio.
* Una vez que la URL del video es devuelta (return(Video)), el sistema puede proceder a reproducir el vídeo (reproducirVideo()).

**15.1.3.1. Control**

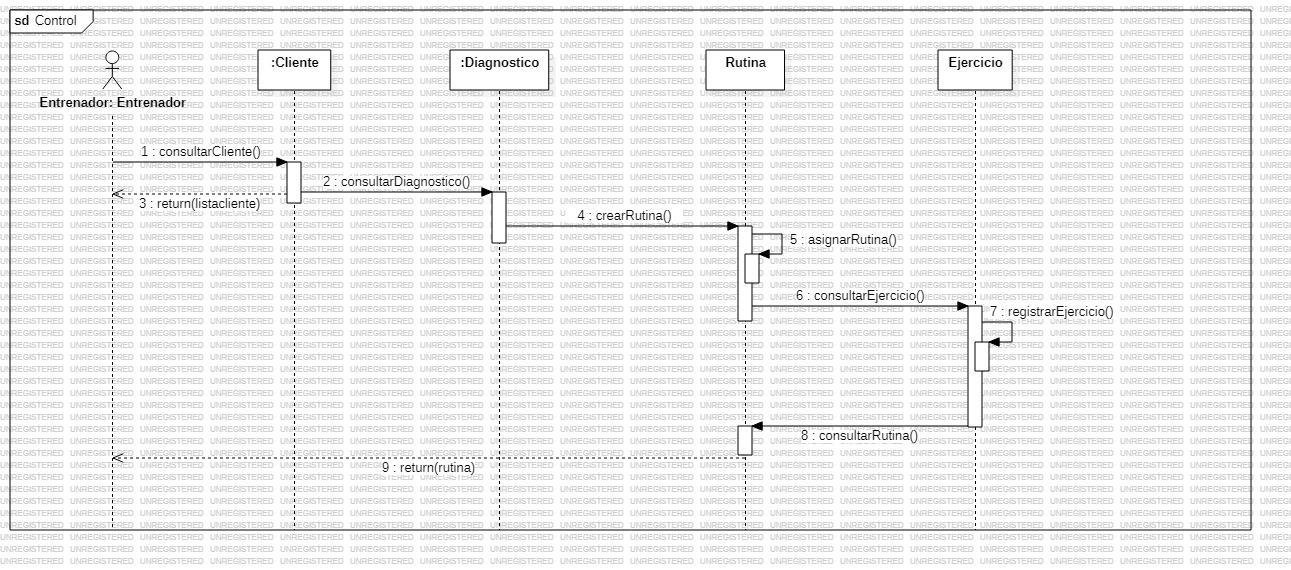


Figura 9 [Diagrama de Secuencias Control](https://drive.google.com/file/d/1dhWI0inV_nxOTEyCRTNs-foJYd7An6-F/view?usp=sharing)

El **diagrama de secuencias** que has proporcionado representa una interacción entre un **Entrenador** y varias clases del sistema: **Cliente**, **Diagnóstico**, **Rutina** y **Ejercicio**. Este diagrama especifica el flujo y orden de los mensajes que se intercambian en un escenario particular, el cual podría ser parte de un sistema de administración de rutinas de ejercicio.

### Flujo del Diagrama de Secuencias:

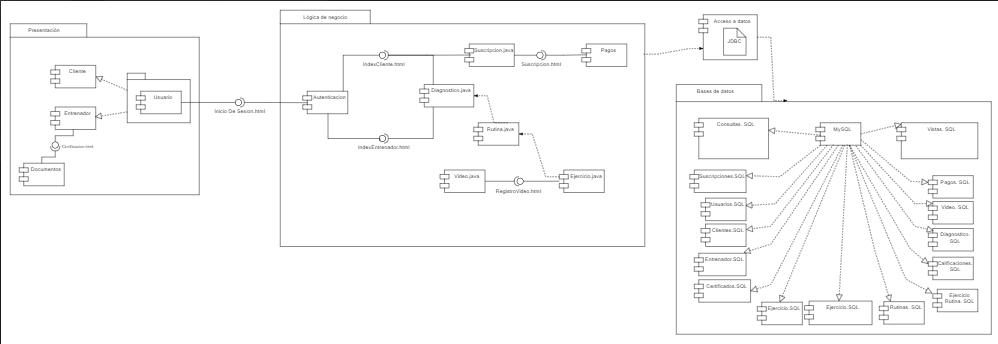
1. **consultarCliente()**
   * El **Entrenador** inicia consultando la lista de clientes disponibles. Este mensaje es enviado al objeto **Cliente**.
   * **Cliente** responde devolviendo una lista de clientes disponibles.
2. **consultarDiagnostico()**
   * El siguiente paso es que el **Entrenador** consulta el diagnóstico del cliente seleccionado. El mensaje se envía al objeto **Diagnostico**.
   * El objeto **Diagnostico** devuelve la información relevante del diagnóstico del cliente.
3. **crearRutina()**
   * Con base en el diagnóstico recibido, el **Entrenador** envía una solicitud al objeto **Rutina** para crear una nueva rutina personalizada para el cliente.
   * **Rutina** no responde inmediatamente; su creación está en curso.
4. **asignarRutina()**
   * Una vez creada la rutina, el **Entrenador** asigna esta rutina al cliente mediante el mensaje **asignarRutina()** enviado nuevamente al objeto **Rutina**.
5. **consultarEjercicio()**
   * El **Entrenador** luego consulta los ejercicios disponibles que pueden formar parte de la rutina. Este mensaje es enviado al objeto **Ejercicio**.
6. **registrarEjercicio()**
   * Tras seleccionar los ejercicios adecuados, el **Entrenador** registra estos ejercicios en el sistema mediante el mensaje **registrarEjercicio()** enviado al objeto **Ejercicio**.
7. **consultarRutina()**
   * Finalmente, el **Entrenador** consulta la rutina completa para verificar que se haya registrado correctamente.

Este diagrama refleja un flujo común en un sistema de gestión de clientes en el ámbito fitness, donde un entrenador personaliza las rutinas basadas en el diagnóstico de un cliente y luego asigna y registra los ejercicios correspondientes.

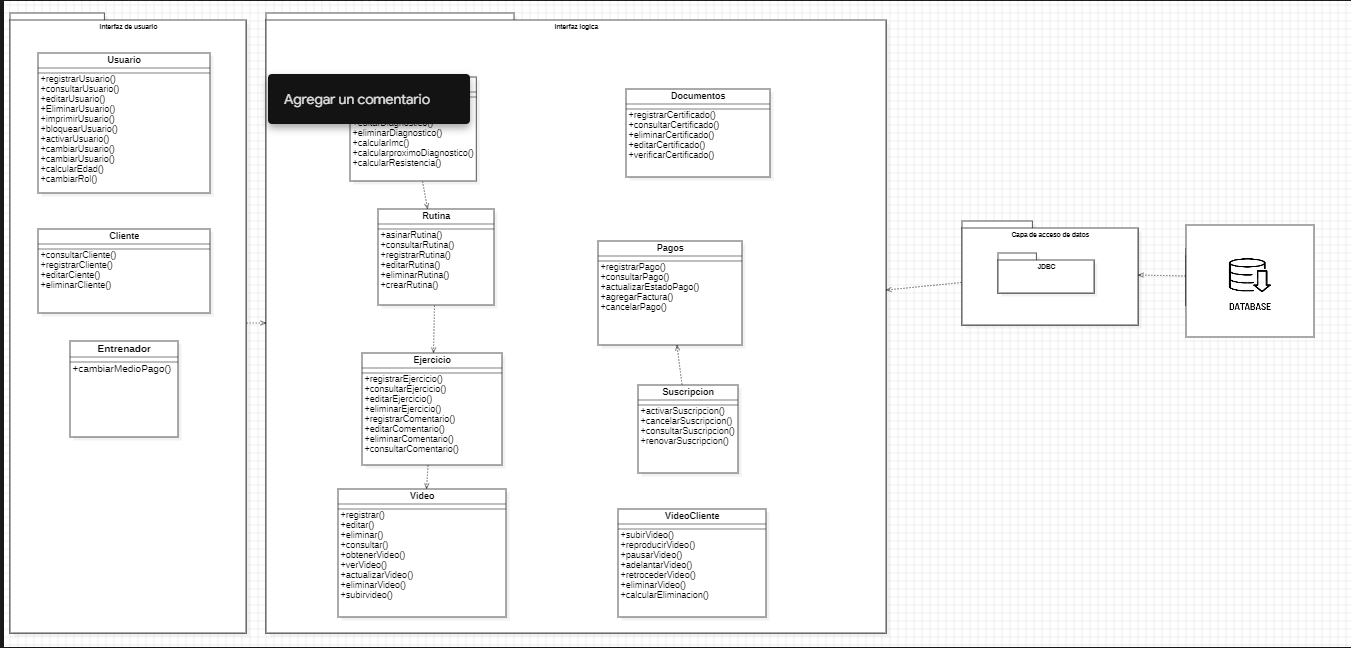
El diagrama de secuencias detalla el orden en que ocurren estas acciones y cómo se comunican los diferentes objetos dentro del sistema para llevar a cabo la funcionalidad deseada.

### 6. Registro de Ejercicios y Rutinas:

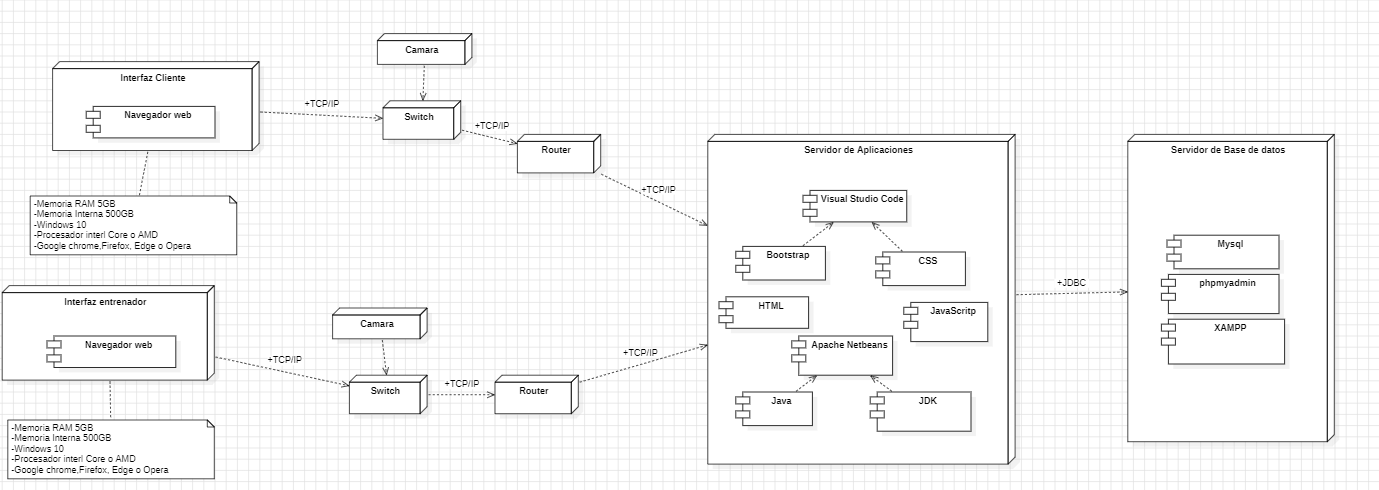
1. El sistema registra los **ejercicios nuevos** agregados a la rutina (registrarEjercicio()).
2. Finalmente, la **rutina** completa, con los ejercicios y videos asociados, se **registra** para el cliente (registrarRutina()).
   1. **VISTA DE IMPLEMENTACIÓN**
      1. **Diagrama de Componentes**

****

* + 1. **Diagrama de Paquetes**

****

* 1. **VISTA DE DESPLIEGUE**
     1. **Diagrama de despliegue**

****

1. **Arquitectura en capas**

| Capa de presentación | * HTML * BOOTSTRAP * JAVASCRIPT * CSS * Visual Studio |
| --- | --- |
| Capa de negocio | * Apache netbeans * Java * jdk |
| Capa de persistencia | * JDBC |
| Capa de datos | * MYSQL * XAMPP * PHPmyadmin |

1. **VISTA DE DATOS**
   1. Modelo Relacional normalizado tercera forma normal de la base de datos (SGBD).
2. **Definición de Interfaces de Usuario**

* Página principal, formularios, páginas error.
* Landing Page (Página Principal).
* Pantallazos interfaces modulares.

1. **Características Generales de Calidad**
   1. **Tamaño y performance**

* Tiempo de respuesta en el acceso a la Base de Datos: 3 seg
* Tiempo de respuesta de transacciones: 5 seg
* Espacio en disco para el cliente: 8 GB
* Espacio en disco para el servidor de Base de datos: 1 Tb

Puntos 7.2. al 7.7., se debe argumentar por cada atributo de calidad, como se cumple el mismo para el Software.

* 1. **Calidad**

La calidad en el software SABI se asegura a través de la implementación de atributos clave que garantizan una experiencia confiable, eficiente y segura. Esto se traduce en:

* Entregar un producto funcional y alineado con los requisitos del usuario.
* Minimizar errores mediante pruebas exhaustivas.
* Satisfacer las expectativas de usabilidad, eficiencia y mantenimiento.

* 1. **Usabilidad**

El software SABI está diseñado con una interfaz intuitiva y amigable, que permite a los usuarios, entrenadores y administradores navegar fácilmente entre sus funciones. La usabilidad se cumple al:

* Incorporar menús claros y opciones accesibles que guíen al usuario en su interacción.
* Permitir personalización de rutinas y reportes para adaptarse a las necesidades individuales de cada cliente.
* Garantizar que los elementos visuales (botones, gráficos y reportes) estén organizados de manera comprensible.
* Ofrecer tutoriales o guías de uso para minimizar el tiempo de aprendizaje.

* 1. **Eficiencia**

SABI utiliza un diseño optimizado en su arquitectura y lógica de negocio para procesar información rápidamente. Esto asegura:

* Respuesta rápida en el registro, consulta y generación de reportes de resultados.
* Eficiencia en la gestión de grandes volúmenes de datos, como rutinas y diagnósticos, a través de consultas SQL optimizadas.

* 1. **Seguridad**

El software prioriza la protección de los datos personales y de progreso de los usuarios al:

* Implementar autenticación mediante contraseñas robustas y encriptadas.
* Establecer controles de acceso diferenciados, permitiendo que solo ciertos roles (administradores o entrenadores) gestionen información sensible.

* 1. **Confiabilidad**

La confiabilidad del software SABI se garantiza al:

* Ofrecer un funcionamiento estable con un registro constante del uso de funciones críticas, como el diagnóstico inicial y la generación de reportes.
* Incluir un sistema de notificaciones y alertas para asegurar que los usuarios cumplan con sus rutinas.
* 1. **Mantenimiento**

El diseño modular de SABI facilita su mantenimiento y actualización al:

* Separar la lógica de negocio, la interfaz de usuario y el acceso a datos en capas claramente definidas.
* Documentar el código y las funcionalidades para que futuros desarrolladores puedan realizar ajustes o mejoras fácilmente.
* Contar con pruebas automatizadas que permiten verificar la estabilidad del sistema tras cambios o nuevas implementaciones.

* 1. **Estándares**

El desarrollo de SABI se alinea con estándares reconocidos para garantizar su calidad:

* Uso de principios de diseño responsivo para que funcione en distintos dispositivos.
* Implementación de buenas prácticas en bases de datos relacionales, como normalización y uso de claves primarias/foráneas.

**21.9** Normas de calidad, mencionar dos normas de calidad de Desarrollo de Software y argumentar cómo esas normas aplican para el software.

### Normas de calidad en el desarrollo de software

* **ISO/IEC 25010:2011 (Modelo de calidad del producto de software)**
* **Argumentación para SABI**: Esta norma define las características que debe tener un software de calidad, incluyendo:
* **Adecuación funcional**: SABI cumple al gestionar y controlar hábitos de vida, cumpliendo con los objetivos planteados.
* **Eficiencia del rendimiento**: Asegura que las funciones, como reportes y diagnósticos, se ejecuten rápidamente.
* **Seguridad**: Protege los datos de los usuarios mediante autenticación y encriptación.
* **Mantenibilidad**: La arquitectura modular y documentada de SABI permite actualizaciones y correcciones eficientes.

Implementar esta norma asegura que el software cumpla con estándares internacionales reconocidos de calidad, alineados con las expectativas del usuario.

**ISO/IEC 12207:2017 (Procesos del ciclo de vida del software)**

* **Argumentación para SABI**: Esta norma establece los procesos necesarios durante el ciclo de vida del desarrollo, incluyendo:
* **Análisis de requisitos**: SABI se basa en un análisis previo para entender las necesidades de los usuarios.
* **Diseño y desarrollo**: La arquitectura en capas asegura claridad y robustez en el diseño.
* **Pruebas y validación**: Garantiza que cada módulo, desde la gestión del diagnóstico hasta los reportes, funcione correctamente antes de ser entregado.
* **Mantenimiento y soporte**: La modularidad de SABI facilita su actualización continua y la corrección de errores.
* Aplicar esta norma asegura que el desarrollo del software sea metódico, cubriendo todas las etapas desde la planeación hasta el mantenimiento.