



Universidad de Las Américas

Facultad de Ingeniería y Ciencias Aplicadas

INGENIERÍA DE SOFTWARE

*Sistema web de gestión de información y desempeño de entrenamientos
utilizando inteligencia artificial para la Academia de Esgrima Ciudad de
Quito*

Diego Hiriart León

Luis Augusto Corales Martínez

Christian Andrés Samaniego Cañizares

01 de junio de 2023

Quito, Ecuador



1. Introducción

1.1. Identificación y descripción del problema o necesidad

La Academia de Esgrima Ciudad de Quito es una institución dedicada la formación en el deporte de esgrima. Debido a la cantidad de atletas que forman parte de la academia, y a los constantes entrenamientos y competencias, es necesario que los entrenadores tengan muy en cuenta las necesidades particulares de preparación y práctica de todos los alumnos. Cada esgrimista tiene diferentes habilidades que reforzar o mantener, y los entrenadores deben poder enfocarse en identificar estos aspectos y registrarlos. Además, los atletas a veces pueden necesitar entrenar solos, o requerir retroalimentación de su desempeño en los entrenamientos; esto en forma de la correctitud de sus técnicas y su desempeño en combates de práctica.

Manejar esta información de manera manual, tanto para entrenadores como esgrimistas, no resulta ideal; ya sea por posibles olvidos o por la gran cantidad de información que se debe considerar para personalizar los entrenamientos. Por otro lado, para mantener registro del desempeño (puntaje) en los combates, no se tiene un registro más allá de la memoria inmediata de los esgrimistas y entrenadores. Esto es porque las máquinas de puntaje existentes en la academia solo indican cuando se hacen toques, mas no mantienen conteo. Al hablar de combates, se está haciendo referencia a enfrentamientos entre atletas con el arma de su preferencia (espada, florete, o sable), en los cuales se busca anotar puntos haciendo toques al oponente con el arma. Se pueden encontrar más detalles del deporte visitando la página de documentos de la Federación Internacional de Esgrima, en el [Anexo 1](#).



Imagen 1 Ejemplo de una máquina de puntaje como las usadas en la Academia, solo marca cuando se hace un punto, no el tiempo o puntaje acumulado (Absolute Fencing Gear, 2022)

udla



Imagen 2 Máquina de puntaje con un módulo integrado para conteo y temporizador (Absolute Fencing Gear, 2022)

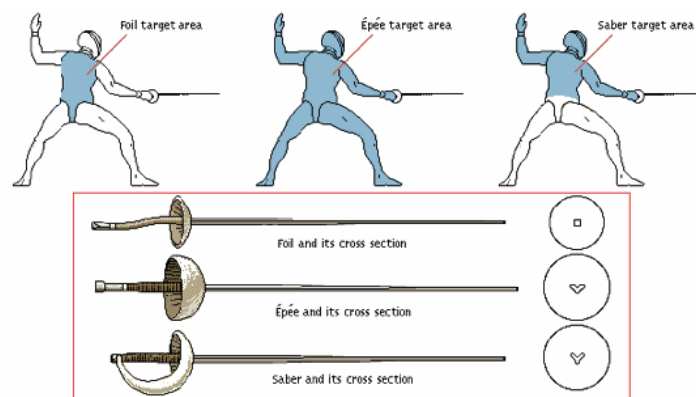


Imagen 3 Diferencias entre las 3 armas de esgrima y los blancos válidos para cada una (Olympia Fencing Center, 2021)



Imagen 4 Combate de espadistas en las Olimpiadas de Tokio 2020

El problema radica en que, aunque los entrenadores recuerdan las áreas que debe reforzar cada esgrimista, ya que gran cantidad de información del desempeño se



registra mentalmente, no hay registro histórico de su desempeño, mejorías o deterioros en sus técnicas, y rendimiento en combates. Al no contar con esta información detallada, les es difícil a los entrenadores y los deportistas estar conscientes de cambios en el desempeño más allá de observaciones instantáneas en los entrenamientos. Esto ocasiona que no se tengan datos acerca del desempeño de cada deportista, solamente apreciaciones periódicas por parte de los entrenadores que no quedan registradas. Tales observaciones son valiosas, pero al no tener un histórico o métricas de estas, es difícil evaluar la evolución de cada deportista a través del tiempo y continuar ajustando los entrenamientos. Como mencionan Apostolou y Tjortjis (2019) contar con datos acerca de los deportistas es de importancia para mejorar tácticas de entrenamiento; Sarlis y Tjortjis (2020) denotan que estos datos son útiles para determinar puntos de mejora, fortalezas y debilidades de atletas. Por otro lado, si los alumnos quisieran evaluar su propio rendimiento para reforzar técnicas o habilidades por su cuenta, deben recurrir siempre personalmente a su entrenador, en lugar de poder revisar datos de sus entrenamientos para enfocarse en mejorías a partir de la retroalimentación.

Considerando estas necesidades, la Academia requiere algún tipo de herramienta que: permita a los entrenadores gestionar los planes para cada alumno, brinde a los esgrimistas información sobre sus planes de entrenamiento, y permita a entrenadores y deportistas seguir el desempeño en los combates de práctica en la Academia.

1.2. Descripción de la empresa

La Academia de Esgrima Ciudad de Quito es una institución ubicada en el Rancho San Vicente del Club de Oficiales de la Policía Nacional, dedicada a la formación de deportistas en la esgrima. Esta institución cuenta con esgrimistas de múltiples rangos de edades, que participan en campeonatos a nivel local, nacional, e internacional. En comparación a otras academias y clubes de esgrima, la Academia de Esgrima Ciudad de Quito es relativamente nueva, empezando a ofrecer sus servicios en el 2017. La Academia cuenta con tres entrenadores, dos de ellos a tiempo completo, los cuales se encargan de la formación de los alumnos en los aspectos relacionado al deporte de esgrima. En la Academia, se practica esgrima principalmente con dos de las tres armas propias del deporte, espada y florete. Se puede revisar el [Anexo 2](#) para visitar la página de la Academia. Organizacionalmente, la academia tiene la siguiente estructura:

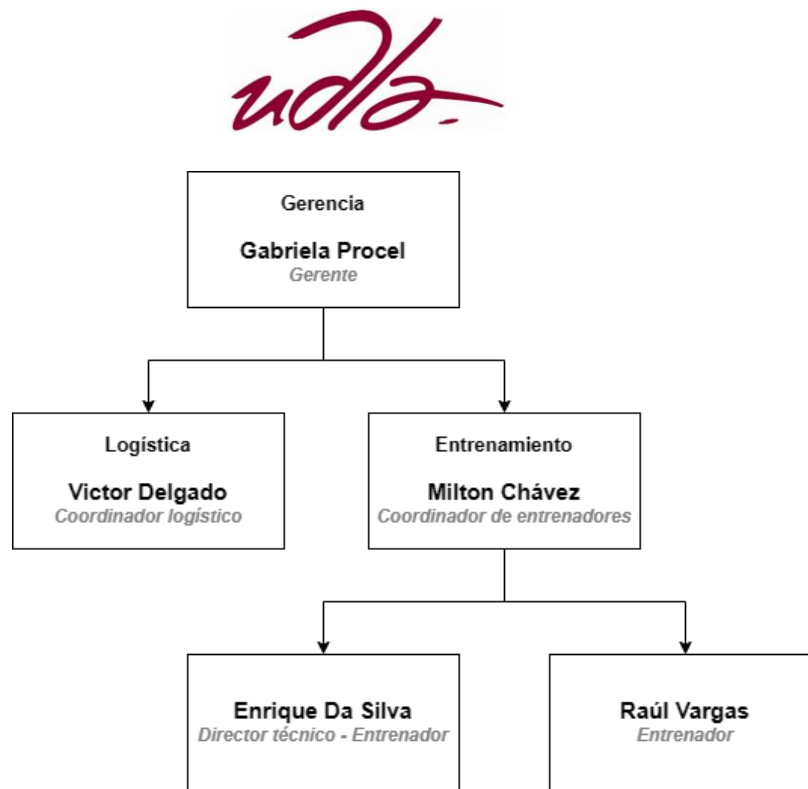


Diagrama 1 Organizagrama de la Academia de Esgrima Ciudad de Quito

Es en el área de entrenamiento de la Academia que se aplicará la solución. Los tres entrenadores aportan con la formación de los esgrimistas, pero tienen especializaciones en armas específicas. El coordinador de entrenadores, Milton Chávez, entrena al grupo de espada y florete; Raúl Vargas es entrenador del grupo de élite de florete de la Academia; y Enrique Da Silva además, de ser entrenador de ambos grupos, es director técnico de la Academia. Existen 3 horarios de entrenamiento en la Academia, en la mañana de 10am a 12 pm, y en la tarde de 4pm a 5:30pm, y de 5pm a 7:30pm; este último es un horario enfocado al grupo de élite de la Academia, involucrado en competencias internacionales con frecuencia. Estos horarios se cumplen de lunes a viernes, y los sábados existe un solo horario de 10am a 12pm.

1.3. Impacto del proyecto en la sociedad

La asistencia que brinda un sistema de este tipo a los esgrimistas y entrenadores permitiría mejorar el rendimiento de los deportistas. No solo existe esta mejoría por la asistencia a actividades de los entrenadores, sino por la ayuda que el sistema brindaría a los deportistas. Al permitirles contar con retroalimentaciones de sus prácticas, pueden usar estos datos para mejorar sus actividades de preparación. Esto, en conjunto a las actividades del entrenador, permite mejores procesos de entrenamiento que resulten en rendimiento elevado.

Por otro lado, el proyecto está orientado a asistir en la gestión de entrenamiento de la Academia de Esgrima Ciudad de Quito, que podría agilizar las actividades de la institución, permitiéndoles ser capaces de gestionar más esgrimistas. Como consecuencia, existirían más practicantes de esgrima en la ciudad, un deporte que no tiene alta popularidad en comparación a otros en el país. Por la existencia de nuevos atletas, la comunidad local de esgrima crecería, lo que a su vez podría resultar en mayor



actividad en competencias y eventos, incluso pudiéndose descubrir personas con talento que representen a sus clubes o al país en competencias locales e internacionales. También al empezar a ejercitarse más personas en la Academia, o con otros clubes por el incremento de popularidad, puede existir un efecto positivo en la salud de las personas por realizar esta actividad de atletismo.

Adicionalmente, esta solución, al estar orientada alrededor de la esgrima, puede ser personalizada para su uso por otros clubes y academias. Este ya es el caso con otras disciplinas deportivas, existen athlete management system (AMS) como Aces para tennis; Kitman Labs usado por ligas de football americano, basketball, hockey, entre otros (G2, 2023); o Athlete Analyzer que se especializa en artes marciales (Athlete Analyzer, 2022). También se utiliza video assistant referee (VAR) en football (FIFA, 2022), para monitorear con video más disputas en partido; un sistema similar existe en esgrima, pero solo como revisiones de grabación en cámara lenta de los combates. Además, si más clubes pueden usar esta solución, se potenciaría más el impacto previamente descrito en la comunidad de esgrima local.

2. Análisis de posibles soluciones

2.1. Identificación y selección de la mejor solución

Tres posibles soluciones se exploraron para el desarrollo del proyecto:

- 1) Sistema de hardware y software que cuente con:
 - Una aplicación web (sitio web) que permita a los entrenadores definir planes personalizados de entrenamiento para cada atleta, y gestionarlos; los atletas podrían consultar estos datos en la misma plataforma.
 - Un componente de inteligencia artificial orientado a asistir a los atletas cuando realicen entrenamiento individual. Usando visión por computadora, se identificarían posibles errores que los esgrimistas pueden estar cometiendo en los movimientos durante entrenamientos los mismos que se podrán visualizar mediante dashboard y diagramas de las poses incorrectas.
 - Máquinas (hardware) de puntaje que permitan llevar registro de los toques en los combates. Tales máquinas actuarían como módulos adicionales a los equipos de puntaje existentes en la Academia.
- 2) Un sitio web que permita a los entrenadores:
 - Definir planes no personalizados (fijos) que los alumnos podrán consultar y seguir.
 - Mantener registro de errores comunes que están visualizando en los entrenamientos de cada alumno, los alumnos pueden acceder a ver esta retroalimentación.
 - Ingresar manualmente los resultados de los combates que se realicen al entrenar, información que también estará disponible para los alumnos.



3) Sistema de software con los siguientes componentes:

- Una aplicación web para smartphones que permita a los entrenadores manejar y personalizar los entrenamientos de los alumnos registrados en la aplicación.
- Un componente de inteligencia artificial que permita grabar un video del entrenamiento del alumno para que seguidamente pueda ser procesado y realizar la retroalimentación y puntos a mejorar para el alumno, la cual será enviada a la aplicación web.
- Desarrollar una aplicación móvil que permita llevar el puntaje de los combates manualmente.

Analizando las posibles soluciones, se encontró que la primera opción, el sistema web con hardware y software, es la solución óptima. Esto debido a que presenta funcionalidades serán más beneficiosas para la Academia, y se ajustan adecuadamente a restricciones de tiempo, costos y experiencia de los integrantes del grupo del proyecto. Para determinar la mejor solución, se consideran tres criterios: costo, complejidad (vinculada estrechamente a tiempo y experiencia), y completitud de la solución (atención a las necesidades y el problema). A continuación, se detalla una comparativa entre las soluciones y cómo esta justifica el escoger la primera opción.

Todas las soluciones representan un gasto necesario para hosting de sitio web y base de datos, se diferencian en el costo de aplicar partes específicas de las soluciones. La primera opción de solución presenta gastos por la construcción de las máquinas para puntaje, este es hardware que debe ser adquirido para cada extensión a crearse; la segunda opción no presenta costos adicionales; y la tercera alternativa representa gastos en licencias para publicar la aplicación móvil en la tienda de Apple. Aunque la segunda alternativa de solución es mejor en este criterio, no es lo mejor respecto a completitud.

En términos de complejidad al implementar las soluciones, la primera y tercera opción presentan complejidad elevada. Esto, debido al desarrollo de máquinas de hardware en la primera opción y desarrollo de aplicaciones móviles para dos sistemas operativos en la tercera; además, ambas cuentan con un módulo de inteligencia artificial. La segunda alternativa de solución es de complejidad baja, pero esto es un reflejo de que no es ideal para resolver correctamente el problema identificado.

El criterio de completitud de las soluciones es el aspecto que más las diferencia. La segunda solución no llega a resolver adecuadamente los problemas identificados, pues no permite personalización de los entrenamientos, un punto muy importante. La tercera opción de solución permite personalización, registro de errores detectados, y seguimiento de desempeño en combates. Sin embargo, esta no es una solución óptima porque, para almacenar puntajes, los esgrimistas deben acudir constantemente a la aplicación móvil o retirar parte de su concentración del combate para recordar el puntaje y luego registrarlo. Por último, la primera solución permite personalización,



detección de probables errores en los entrenamientos, y registro de puntajes resultantes en los combates de entrenamiento mediante una máquina que se puede consultar al finalizar el combate para registrar el resultado en la aplicación web de forma más sencilla.

Criterio / alternativa	Solución 1	Solución 2	Solución 3
Costo	Medio	Bajo	Medio
Complejidad	Alto	Bajo	Alto
Complejidad de la solución	Alto	Bajo	Medio

Tabla 1 Comparativa de soluciones según costo, complejidad, y completitud de la solución

Considerando la comparación y evaluación realizada a las 3 alternativas de solución, se determinó que la primera opción es la ideal para resolver el problema identificado en la Academia. Esto se debe a que, a pesar de presentar un costo más alto que la alternativa número dos, su complejidad resulta en una atención más completa a los problemas identificados. Aunque en este aspecto de completitud la tercera alternativa es muy acertada también, la manera de registrar puntajes resulta inconveniente para los esgrimistas, pues deberían concentrarse en sus combates y no en revisar constantemente sus dispositivos móviles para registrar puntos. Todo esto define a la primera solución como la ideal para el proyecto, pues a pesar de su costo y complejidad atiende las necesidades de la Academia completamente y de manera más cómoda para los esgrimistas.

3. Objetivos

3.1. Objetivo General

Desarrollar una aplicación web para la Academia de Esgrima Ciudad de Quito, la cual permita la gestión de datos de entrenamiento, la aplicación se apoyará de una inteligencia artificial para detección de posibles errores en entrenamientos, y una máquina de conteo de puntaje para combates.

3.2. Objetivos Específicos

- Realizar una aplicación web, para la gestión de planes de entrenamiento personales con carga y consulta de datos de rendimiento y combates.
- Integrar a la aplicación web un módulo de visión por computadora para retroalimentación a ejercicios individuales de esgrima.
- Ensamblar y programar un dispositivo electrónico para llevar el conteo de puntos en combates de esgrima.

4. Alcance

4.1. Alcance de la solución seleccionada

El proyecto consiste en tres partes principales, que conforman el sistema web para asistir en la gestión de desempeño e información de entrenamientos. Los entrenadores



y alumnos podrán usar estos componentes para consulta y registro de datos. A continuación, se detalla el alcance del proyecto.

Primero, la aplicación web para la gestión de datos de entrenamiento. Esta aplicación cuenta con dos módulos, cada uno con funcionalidades específicas para atender el problema. Además, esta aplicación consiste en tres entregables para ponerla en funcionamiento.

- Módulo para entrenadores. En esta parte de la aplicación los entrenadores podrán:
 - Registrar alumnos y gestionar sus datos.
 - Crear y actualizar planes de entrenamiento.
 - Registrar resultados de combates de cualquier par de alumnos.
 - Consultar posibles errores detectados mediante dashboard y diagramas de las poses erróneas.
 - Visualizar datos de rendimiento en combates.
 - Gestionar información de retroalimentación acerca del desempeño de los alumnos en base a observaciones y datos de combates y posibles errores.
- Módulo para alumnos. Para facilitar la consulta de datos por parte de alumnos y registro de combates, permite:
 - Registrar datos de combate, solo de aquellos enfrentamientos en los que hayan participado personalmente.
 - Utilizar la funcionalidad de detección de potenciales errores en video por inteligencia artificial.
 - Acceder a sus planes de entrenamiento y retroalimentaciones.
 - Consultar datos de desempeño en combates.
 - Consultar información de detección de posibles errores mediante dashboard y diagramas de las poses incorrectas realizadas.
- Los entregables de esta aplicación web son:
 - API (backend) para la aplicación web: es decir la lógica de negocio para la gestión de datos de entrenamiento, incluyendo potenciales errores detectados y puntajes.
 - Base de datos para la solución: para almacenamiento de usuarios y datos de entrenamientos y desempeño.
 - La página web (frontend): interfaz de usuario (capa de presentación) para interactuar con el sistema a través de un navegador, permite usar todas las funcionalidades descritas en los módulos.

Segundo, un componente para la identificación de probables errores comunes en las acciones de los entrenamientos. Esta parte del sistema consistirá en una inteligencia artificial que realice reconocimiento de posturas en video, para identificar potenciales errores que se puedan cometer en las acciones (movimientos) al entrenar. Se accede a



esta funcionalidad desde la aplicación web, la cámara del dispositivo donde se la use captura video del esgrimista y analiza cuadros (o frames) del mismo para detectar posibles errores, y almacena los datos recolectados. Este módulo cumple dos funciones principales y le corresponden tres entregables.

- Funciones:
 - Detecta los posibles errores que comentan los alumnos en prácticas individuales.
 - Almacena datos de probables errores detectados para que puedan visualizarse en la interfaz de la aplicación web a manera de dashboard y diagramas de las posiciones erróneas.
- Entregables:
 - Datos de entrenamiento (ejemplos de errores) para la inteligencia artificial: dataset para alimentar de información a la inteligencia artificial, de tal manera que aprenda a detectar los errores requeridos.
 - Modelo de inteligencia artificial para reconocimiento de probables errores: es la inteligencia artificial en sí, que hará el reconocimiento según el entrenamiento proporcionado.
 - Integración con aplicación web: código necesario para enviar datos de detección a la base de datos y visualizarlos en la aplicación web.

Desde la aplicación, el esgrimista podrá activar la funcionalidad de detección, para lo cual debe primero colocar el dispositivo que esté usando de tal manera que capture sus movimientos. Entonces, la aplicación empezará a capturar video por un periodo de tiempo, analizando los frames (cuadros) del video para encontrar posibles errores. Cuando finalice el tiempo de detección los posibles errores identificados se guardarán, y tantos entrenadores como atletas podrán acceder a la consulta de datos de potenciales errores detectados en el dashboard y diagramas de poses incorrectas. Los errores detectados no se visualizarán en tiempo real, para enfocar la concentración del deportista en la práctica a realizarse en lugar de la pantalla del dispositivo.

Por último, una máquina de puntaje para registrar los puntos que se realicen en combates de entrenamiento. La máquina tiene dos funciones principales y dos entregables para su funcionamiento.

- Funcionalidades:
 - Mantener conteo de los puntajes de cada esgrimista en un display, se modificará manualmente el puntaje al hacerse un toque.
 - Temporizador para los combates.



- Entregables de la máquina:
 - Software de control: es el software para el conteo de los puntos y manejo de inputs del usuario.
 - Hardware ensamblado: es decir las máquinas en sí, en las cuales está cargado el software de control desarrollado.

Una vez encendido el hardware, el entrenador o arbitro pueden configurar e iniciar el temporizador. Entonces, los esgrimistas pueden proceder con el combate de manera muy similar a la habitual. Cuando exista un toque y los atletas vuelvan a sus marcas de guardia en la pista, aprovechan esta pausa para aumentar el conteo en la máquina; si un entrenador o arbitro está presente, también lo puede hacer, además de controlar la pausa, continuación, y modificación del temporizador. De esta manera, se evita que tengan que depender de su memoria y diferir su concentración, pues estos datos los mantiene la máquina. Al acabar el combate, uno de los esgrimistas o el entrenador puede registrar el puntaje final en la aplicación web.

4.2. Limitaciones y restricciones del proyecto

2. .

2.2.1. Limitaciones

El alcance del proyecto está delimitado por algunos puntos. A continuación, se describen las limitaciones que se han definido para cada componente del sistema a desarrollar.

- Inteligencia artificial para reconocimiento de posibles errores
 - No se harán identificaciones de tantos errores en acciones compuestas como en acciones simples (acciones básicas de esgrima). Esto se debe a la alta complejidad de implementación que implicaría esto. Las acciones compuestas tienen movimientos mucho más complejos que las simples; por lo tanto, independientemente del tipo de acción, la inteligencia artificial se concentrará en errores de acciones simples o errores comunes para ambos tipos de acciones.
 - La inteligencia artificial solo se puede usar para un esgrimista a la vez. Esto se debe a que el reconocimiento de posibles errores no hará distinciones entre dos esgrimistas distintos. Por esto, no se podría usar este componente del sistema en un combate, pues observaría errores en ambos esgrimistas sin distinción de quien los comete. Lograr la distinción entre dos personas distintas en un combate requeriría desarrollo dedicado de una inteligencia artificial.
- Máquinas de puntaje
 - Debe ser posible subir y bajar el conteo de puntaje se de manera manual. Esto se debe a que, para controlar que el puntaje aumente automáticamente, las máquinas tendrían que conectarse al cableado interno de los equipos de la Academia. Hacer esto significaría realizar



ingeniería inversa en el hardware existente para saber dónde hacer las conexiones, lo cual sale del enfoque de software que tiene el proyecto. Además, se correría el riesgo de causar daños en los dispositivos de la Academia.

- Se debe poder realizar la carga de datos de puntaje manualmente, la máquina se encargará de mantener el conteo y tiempo, pero el usuario debe acceder a la funcionalidad en la aplicación web al acabar un combate y registrar la información. Esto porque la mejor manera de saber cuándo se ha acabado el combate y se deben registrar los puntos, es el criterio humano.
- La máquina, en concordancia a las regulaciones más actuales de la FIE debe, en los últimos 10 segundos de cada tiempo del combate, mostrar el tiempo testante con decimas y centésimas de segundo.

2.2.2. Supuestos

Para que el proyecto puede desarrollarse de manera ideal y se pueda trabajar para cumplir los objetivos de este tal como se lo tiene planeado actualmente, se deben cumplir algunos supuestos. Estos supuestos se enuncian a continuación con una clasificación simple.

- Supuestos de costos
 - La tecnología, tanto software como hardware y plataformas en la nube, requerida para los componentes del sistema continúan disponibles al costo que actualmente se tiene previsto. Algunas tecnologías son de libre acceso, como las librerías de inteligencia artificial, y otras tienen un costo, como el hardware de las máquinas de puntaje. Esencialmente, se requiere que estos costos no suban, o si lo hacen que no sea drásticamente y sea aceptable para la Academia y los integrantes del proyecto.
- Supuestos de datos e información
 - Para el entrenamiento de la inteligencia artificial es necesario recopilar gran cantidad de datos, en forma de imágenes y videos. Si estos datos no están disponibles en la cantidad requerida, no se podrá entrenar a la inteligencia artificial lo suficiente como tener la precisión adecuada al detectar potenciales errores.
- Supuestos de tiempo
 - No existen retrasos o cambios mayores en el cronograma de actividades para el proyecto que plantea la UDLA.
 - No se dan retrasos significativos en la comunicación de rechazos o aprobaciones a las propuestas que deba revisar la UDLA o la Academia.
- Tecnológicos
 - El acceso a la tecnología (software, hardware, y plataformas en la nube) se mantiene para el país o región. Independientemente de su costo, este supuesto se refiere a que, por ejemplo, que el hardware requerido aún sea



adquirible en Ecuador, o que no se empiecen a implementar restricciones de acceso al software o plataformas para la región.

2.2.3. Restricciones

Siguiendo las 3 categorías del triángulo de restricciones, se tienen:

- Costo:
 - El costo del hardware requerido y el hosting web necesitado deben ser aprobados por la Academia.
 - Actualmente, se tiene presupuestado para el hardware de cada extensión alrededor de \$60 USD. El costo final de hardware puede variar si se ensamblan más máquinas.
 - El costo del hosting está presupuestado para alrededor de \$15 USD mensuales.
 - No existe un costo inicial de implementación para la aplicación web, cómo por ejemplo licencias. Sin embargo, el costo del hosting puede incrementar si se requiere más espacio de almacenamiento para los datos de la aplicación.
- Alcance:
 - Los 3 componentes, con las exclusiones descritas, son los que se desarrollarán en el proyecto. No sería posible extender el alcance sin contar con más tiempo y/o presupuesto.
- Tiempo
 - Para desarrollar el proyecto, se tiene una restricción de cerca de 6 meses, que es el tiempo disponible hasta la presentación del proyecto.

5. Planificación y costos del proyecto

5.1. Estructura de desglose de trabajo (EDT)

Mediante la elaboración de un diagrama EDT (estructura de desglose de trabajo) se identificaron los grupos de actividades y actividades individuales (paquetes de trabajo).

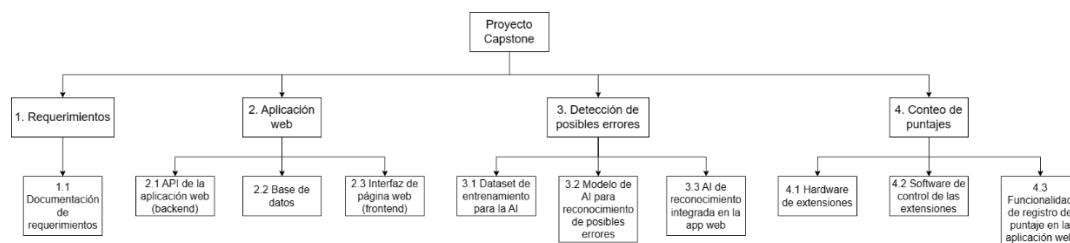


Imagen 5 Diagrama de estructura de desglose de trabajo (EDT)

5.2. Descripción de paquetes de trabajo

Los paquetes de trabajo identificados para la EDT se describen a continuación:



1. Requerimientos

1.1. Documentación de requerimientos

Son los documentos en los cuales constan todas las especificaciones del sistema, obtenidas tras recopilar los requerimientos de manera conjunta con representantes de la Academia.

2. Aplicación web

2.1. API de la aplicación web (backend)

La API que se usará para la aplicación web, esencialmente es la capa de negocio en este caso. Permite conectar la página web con la base de datos y ejecutar cualquier lógica necesaria.

2.2. Base de datos

Capa de persistencia del sistema, contiene todos los datos que se almacenen de usuarios, entrenamientos y desempeño.

2.3. Interfaz de página web (frontend)

Interfaz de usuario de la aplicación web, es la capa de presentación de la aplicación web.

3. Detección de posibles errores

3.1. Dataset de entrenamiento para la AI

Datos para entrenar a la inteligencia artificial para que realice el reconocimiento de errores requeridos. Es un conjunto de datos etiquetados con los cuales el modelo aprende a detectar las acciones erróneas que se desean identificar.

3.2. Modelo de AI para reconocimiento de posibles errores

Es la inteligencia artificial en sí, el modelo que, una vez entrenado, reconocerá los potenciales errores que cometan los deportistas.

3.3. AI de reconocimiento integrada en la app web

La inteligencia artificial, además de ser parte del sistema integrándose en la capa lógica de la aplicación web, debe almacenar los datos de reconocimiento de potenciales errores para que se puedan consultar.

4. Conteo de puntajes

4.1. Hardware

Las máquinas para desarrollarse, al ser en parte hardware, consisten en un microcontrolador y otros componentes como luces y módulos de comunicación que deben ensamblarse.

4.2. Software de máquinas

El microcontrolador de las máquinas requiere contar con firmware que debe ser desarrollado para controlar los componentes de hardware.



4.3. Funcionalidad de registro de puntaje en aplicación web

Para usar los puntajes que se registren en las máquinas de conteo, es necesario contar con un endpoint en la API (backend), es decir la función adecuada en la capa de negocio, junto con la interfaz necesaria en la página web para cargar los puntajes.

5.3. Diagrama de Gantt

En el diagrama de Gantt que se muestra a continuación, se está tomando la duración media de las actividades según PERT (estimación por tres valores), como duración de cada actividad. La descripción de las actividades y valores para estimación se pueden encontrar en el [Anexo 3](#). Además, para observar la planificación con más detalle, se puede usar el archivo de Microsoft Project del [Anexo 4](#).

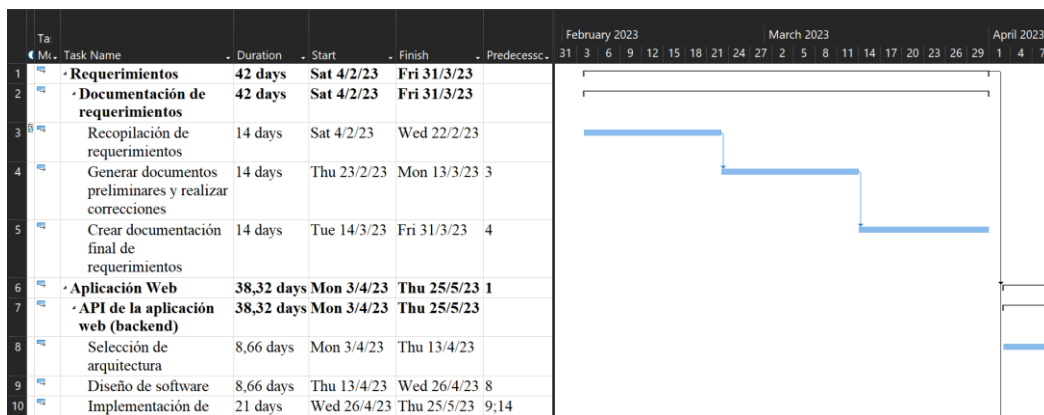


Diagrama 2 Planificación en Diagrama de Gantt 1/6

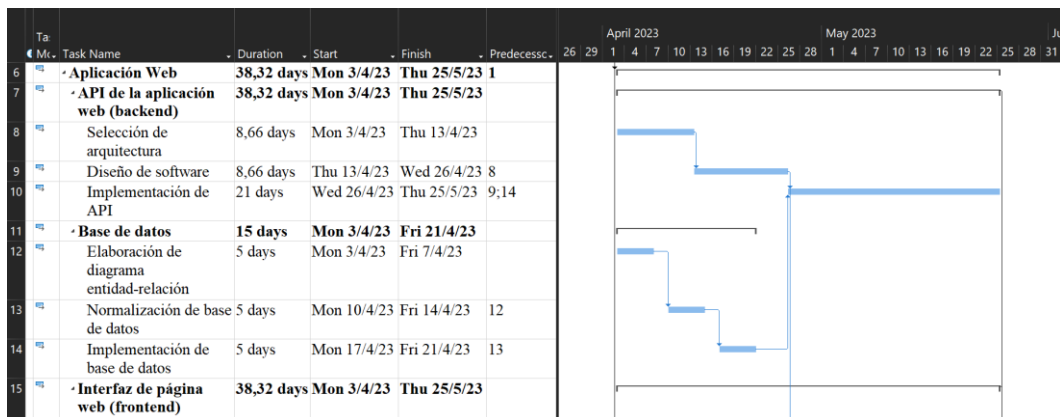


Diagrama 3 Planificación en Diagrama de Gantt 2/6

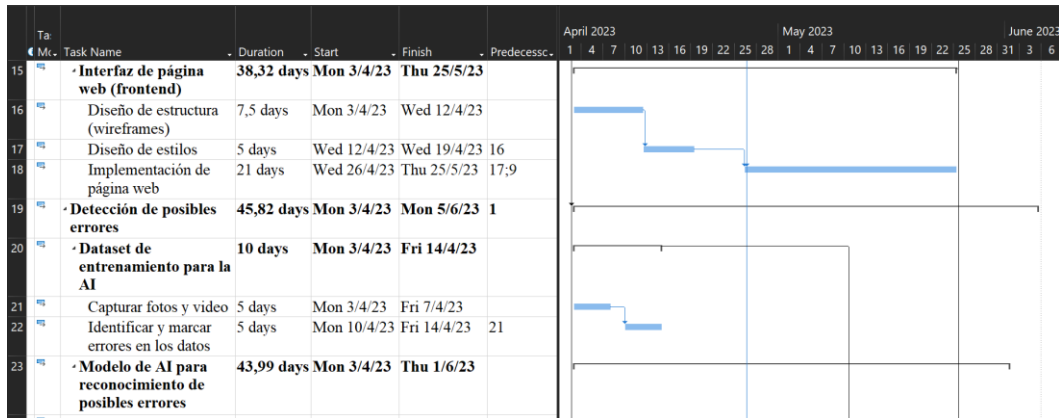


Diagrama 4 Planificación en Diagrama de Gantt 3/6

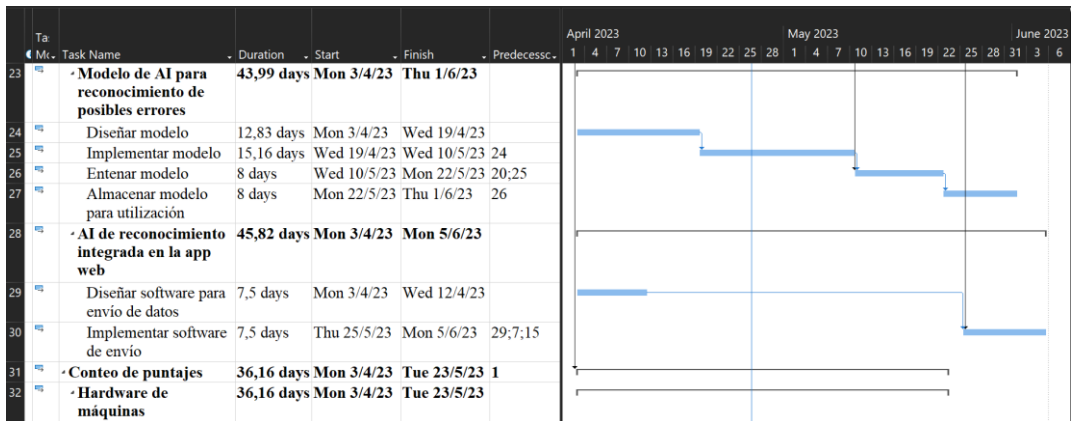


Diagrama 5 Planificación en Diagrama de Gantt 4/6

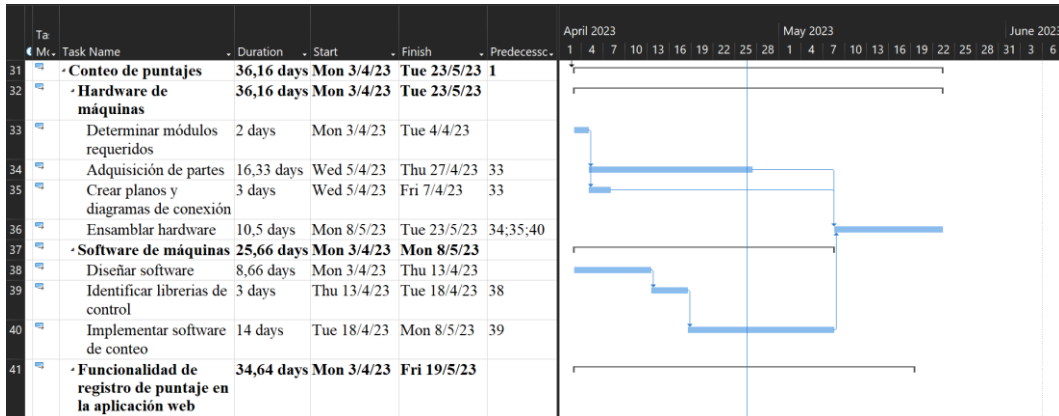


Diagrama 6 Planificación en Diagrama de Gantt 5/6

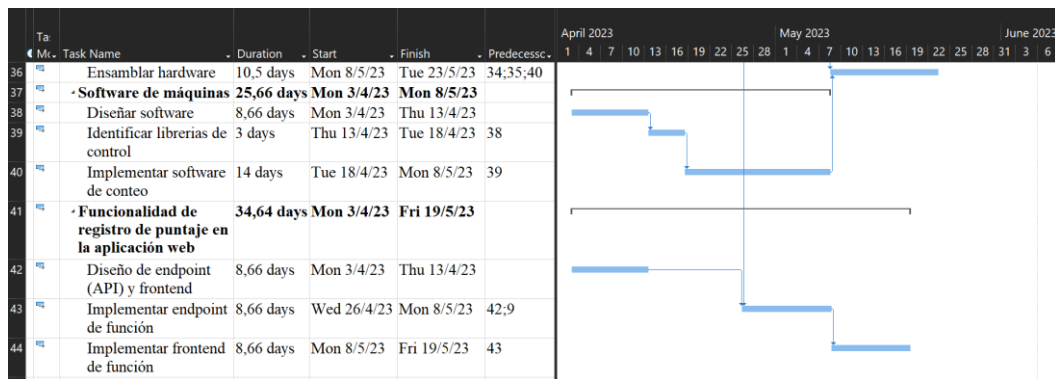


Diagrama 7 Planificación en Diagrama de Gantt 6/6

5.4. Análisis de costos de la solución

Existen tres categorías principales que comprenden costos que involucrará este proyecto, el hosting de la aplicación web, adquisición de componentes de hardware para las máquinas de registro de puntaje, y talento humano. Los pagos que se deben realizar por hosting son mensuales, por tanto, se los considera costos operativos. En cambio, el costo del hardware para las máquinas corresponde al costo de una sola instancia; entonces, junto a costos de desarrollo se le considera gasto en capital. A continuación, se describen estas tres categorías principales, y se muestran los gastos operacionales y de capital con sus componentes en tablas.

- **Hosting:** la aplicación web tendrá tráfico bajo, pues solo será usado por miembros de la academia y no necesariamente de manera concurrente. Por esto, se ha optado por usar el servicio de hosting básico en máquinas virtuales que ofrece DigitalOcean.
- **Hardware:** se requieren algunos componentes de hardware para las máquinas a desarrollarse. Para algunos componentes, se están considerando los precios de importar los mismos, debido a que la diferencia de precio en el país es significativa; por esto también se considera el costo de la importación y peso de los ítems.
- **Recurso humano:** los integrantes del proyecto, es decir quienes desarrollarán el sistema, y un miembro de la Academia quien colabore con la captura de datos de entrenamiento mediante ejercicios erróneos; forman parte del recurso humano para el proyecto.

Gasto de capital		
Item	Descripción	Costo
Microntrolador Arduino MEGA 2560 REV3	Placa de microcontrolador ATmega2560 de 8 bits, 0.13 libras	\$55
Matriz de LEDs	LEDs en formato de matriz para display de conteos, 0.3 libras	\$15



Pulsadores	Botones para interactuar con las máquinas	\$5
Costo de importación	Por libra o fracción \$4.60 USD, más \$10 USD nacionalización	\$14.60
Compensaciones integrantes del proyecto	Salario básico unificado para 3 personas por 5 meses	\$6750.00
Compensación asistente esgrimista para recolección de datos	Colaboración durante 5 días de recolección de fotos y video	\$112.50
Total	-	\$6952.10

Tabla 2 Desglose de gasto de capital

Gasto operacional			
Item	Descripción	Periodicidad	Costo
Hosting	VM Ubuntu, 1 núcleo CPU, 2GB RAM, 50 GB SSD	Mensual	\$12
Total	-	Mensual	\$12

Tabla 3 Desglose de gasto operacional

6. Descripción de estudios realizados

Realizando una búsqueda exhaustiva de proyectos similares en su totalidad o con módulos que se asemejen a lo que se ha propuesto, se han encontrado las siguientes aplicaciones de interés:

- Zenia App: una aplicación móvil para entrenamientos de yoga asistidos con inteligencia artificial, específicamente visión artificial para detectar poses humanas y guiar al usuario con las posturas. Esta aplicación no posee entrenamientos personalizados según las necesidades del usuario, pero si consigue tener una variedad amplia de entrenamientos y provee retroalimentación inmediata al usuario gracias al uso de la inteligencia artificial. (Zenia, 2022).
- CoachUp: una aplicación móvil para entrenamientos deportivos que permite al usuario conectar con entrenadores de todo el mundo (de cualquier deporte) para conseguir entrenamientos personalizados según lo que necesiten. La aplicación no posee un tracking de los entrenamientos ni retroalimentación sobre ellos, ya que solo sirve para entrar en contacto con los entrenadores y planificar los ejercicios con ellos. (CoachUp, 2023).
- Flexit: una aplicación móvil capaz de planificar una serie de entrenamientos personalizados con videos de cómo realizarlos, con tips y guías visuales (como mover el brazo, la postura de las piernas, etc.). Sin embargo, esta aplicación no permite tener una retroalimentación para el usuario, solo lo guía para que realice un entrenamiento específico. (FlexIt Fitness, 2023).



- Sistemas de gestión de atletas: existen sistemas y aplicaciones tipo athlete management system (AMS), como el sistema de Kitman Labs implementado en deportes como basketball, hockey, y football americano; Aces usado para tennis (G2, 2023); y Athlete Analyzer que se especializa gestión de atletas para artes marciales (Athlete Analyzer, 2022).

Sobre el tema de inteligencia artificial capaz de detectar movimientos y/o poses humanas, se han realizado investigaciones con el fin de encontrar los mejores métodos para optimizar los modelos de entrenamiento. En el artículo “Aerobics posture recognition based on neural network and sensors” (Liu, 2022), se evalúa si se pueden usar redes neuronales profundas para reconocer posturas en aeróbicos utilizando video. Como resultado, el estudio muestra que con esta técnica de inteligencia artificial se pueden reconocer adecuadamente poses humanas. Por otro lado, Muñoz (2021) en su estudio “Desarrollo y validación de un sistema sin marcadores para el análisis del movimiento humano” se enfoca en analizar la aplicabilidad de inteligencia artificial para detección de variables como rangos de movimiento de articulaciones. El estudio encontró que las redes neuronales efectivamente pueden ser de utilidad para estimar rangos articulares. La importancia de estos estudios es que sirve como base para el proyecto; ya que no solo compila las mejores prácticas que se podrían usar para el desarrollo propuesto, sino que también ratifica que el uso de una inteligencia artificial es apropiado para capturar movimientos corporales, como los realizados por esgrimistas durante entrenamiento.

7. Desarrollo de la solución

7.1. Diseño de la solución

7.1.1. Requerimientos

Los requerimientos de la solución se recopilaban a lo largo de varias reuniones con los entrenadores de la Academia. En dichas reuniones, se determinaron las funcionalidades que debe cumplir la aplicación web para ajustarse a las necesidades de gestión de entrenamiento, los errores que se desea detectar con la inteligencia artificial, y la configuración y reglas del deporte que cumplir con la máquina para conteo y temporizador. Siguiendo una metodología ágil de desarrollo (explicado más adelante), también se hicieron ajustes a los requerimientos conforme se tenían reuniones de revisión con los clientes.

Para determinar de manera ideal el flujo de las funcionalidades de la aplicación web (incluida la AI) y la máquina, se elaboraron casos de uso detallados en los que se describe el funcionamiento esperado de cada módulo; estos casos se detallan a continuación.

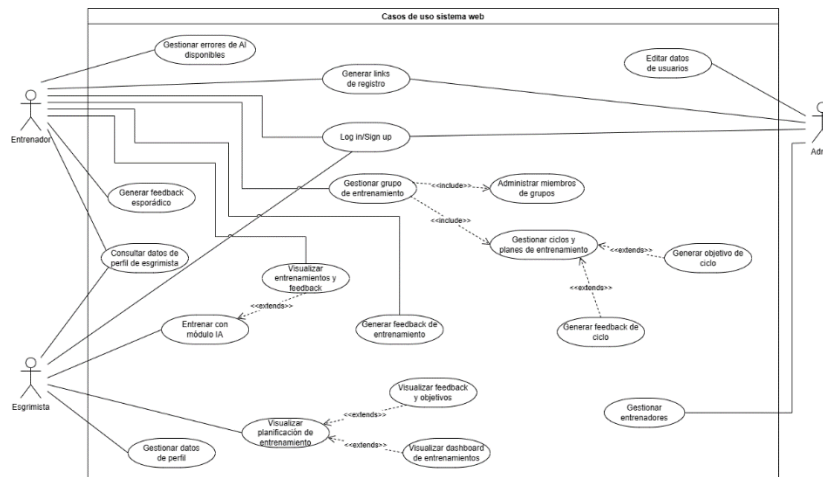


Diagrama 8 Casos de uso para app web

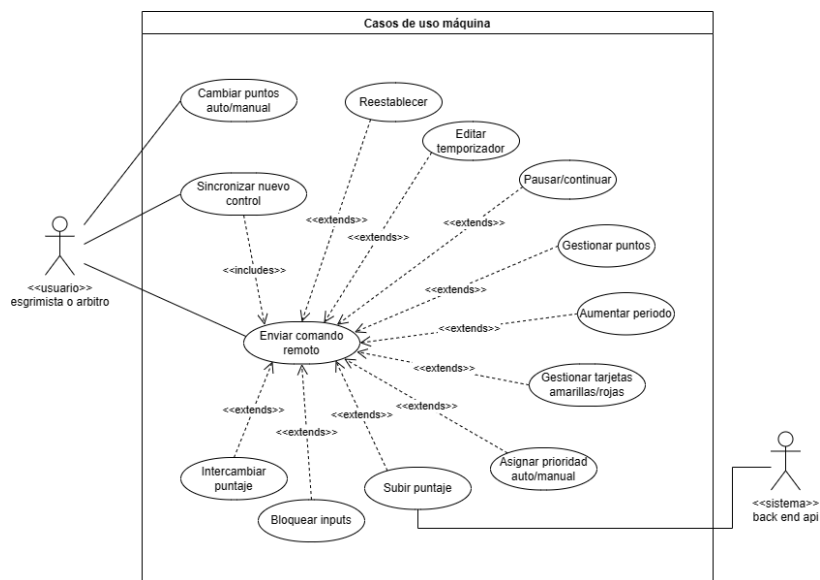


Diagrama 9 Casos de uso para máquina de conteo

7.1.2. Diseño y arquitectura

La solución implementada, de acuerdo con lo descrito en la *sección 2.1* de este documento, consta de: una aplicación web para la gestión de esgrimistas, la planificación de entrenamientos y actividades, y registro de combates; una inteligencia artificial integrada en la aplicación web para detectar posibles errores al entrenar; y una máquina para conteo de puntaje y tiempo. La aplicación web debe ser accesible desde cualquier dispositivo con navegadores que soporten ECMAScript (o JavaScript), considerando que hay distintos tamaños de pantalla para varios dispositivos. Respecto a la máquina a desarrollarse, esta será usada únicamente en las instalaciones de la Academia de Esgrima Ciudad de Quito, pero debe ser capaz de comunicarse con la aplicación web y cumplir ciertas normas de la FIE. A continuación, se detalla cómo la solución está ideada a nivel arquitectónico según estas consideraciones generales y otras más específicas.



Tanto el front end como el back end de la aplicación web se construirán siguiendo una arquitectura por capas. De esta manera se obtiene mayor facilidad de mantenimiento dado que se establece una clara separación de preocupaciones, y se reduce el nivel de acoplamiento entre tecnologías. Por otro lado, es totalmente necesario tener una separación clara de responsabilidades de lógica de negocio y presentación para que, independientemente del dispositivo o interfaz con que se use la aplicación, se pueda acceder a las funciones necesarias de lógica de negocio. Esto también permite trabajar de manera sincrónica y cooperativa desarrollando el back y front end de la aplicación y asegurar compatibilidad.

La capa de negocio (back end) implementa una capa de controladores, los cuales son los encargados de hacer uso del módulo de data y gestionar las respuestas del servidor. El back end se desarrolla un Restful API que permite la entrada de peticiones tanto de información en la base de datos como la ejecución de algoritmos, y esta información será devuelta en formato JSON desde el cliente donde se haya hecho la petición.

Además de poder adaptarse a distintos dispositivos, la aplicación debe ser ajustada correctamente a los tipos de usuario y requerimientos de cada uno. Siguiendo las descripciones de los requerimientos recolectados con la Academia, algunas adecuaciones se hicieron al diseño de esta solución. Esto claramente implica que la aplicación web debe contar con múltiples interfaces para mostrar y recibir distintos tipos de información, así como para cumplir sus funcionalidades. Por esto, se optó por usar el patrón de diseño Single Page Application (SPA) para el front end. Usando SPA, se brinda una experiencia más fluida a los usuarios, y se reduce la complejidad en el lado del servidor. Es de gran importancia, también, que el front end sea totalmente *responsive*. Así, los entrenadores y esgrimistas podrán acceder a gestionar o visualizar información importante desde la comodidad de un dispositivo móvil mientras entrenan, o en las interfaces grandes de un escritorio.

La capa de presentación (front end) es la encargada de renderizar los componentes de forma dinámica con la información obtenida desde la capa de negocio. Después de la carga inicial del documento HTML solamente se realizan actualizaciones parciales en el documento. Para la creación de los componentes de interfaz de usuario se hará uso de la librería React.

El módulo de inteligencia artificial, a pesar de ser parte de la aplicación web, debe ser visualizado esencialmente como un módulo aparte. Aunque se integra con el sistema almacenando los resultados de las sesiones de entrenamiento, es un componente que requiere especial atención para funcionar eficaz y eficientemente. Para la detección y análisis de poses, se tienen componentes de este módulo en el front y back end. Para integrar inteligencia artificial en esta aplicación de ECMAScript, será necesario utilizar o desarrollar una plataforma adecuada, que



pueda correr en el entorno de JavaScript y tener funcionalidades de inteligencia artificial.

La captura de las poses en sí (es decir, la posición del esgrimista) se realizará en el front end, por lo que la inteligencia artificial correrá en el dispositivo en que se esté usando la función. Existen modelos pre entrenados que se pueden utilizar para este propósito, lo importante es que puedan capturar suficientes partes del cuerpo en cada pose. También se debe considerar es que el modelo debe correr en el dispositivo lo suficientemente rápido para capturar suficientes veces las poses en un segundo. Al menos unas 10 detecciones de pose deben realizarse en un segundo para que la inteligencia artificial del back end pueda analizar las mismas y detectar un posible error.

Una vez obtenidas suficientes capturas de poses (esto según la duración del error), entraría en funcionamiento la inteligencia artificial de análisis del back end. Este modelo debe ser construido y entrenado, pues no existe uno al que se pueda simplemente aplicar transfer learning. El modelo a construirse debe ser una red neuronal recurrente, este tipo de red neuronal es necesario porque permite analizar datos teniendo en cuenta la información que se analizó anteriormente.

Para la máquina de conteo de puntaje y temporizador, las consideraciones más importantes, como se ha mencionado, parten de cumplir con las normas de la FIE aplicables además de los requerimientos de la Academia. En lo que respecta a reglamentos, la máquina debe poder ser controlada remotamente; y el temporizador debe mostrar, en los últimos 10 segundos, el tiempo con precisión en décimas de segundo y con centésimas si la máquina está pausada (FIE, 2022). Los requerimientos de la Academia se han basado en tener las funcionalidades básicas con las que cuentan que máquinas oficiales, mismas que se han detallado anteriormente.

La máquina es una combinación de microcontroladores, LEDs para visualización, alertas sonoras, un control remoto, y electrónica básica para interfaz con la máquina de esgrima existente en la Academia. Se reciben las entradas de la máquina existente con algunos componentes electrónicos, mismos que son procesados por un microcontrolador que también acepta comandos de un control remoto (basado en microcontrolador). Con esta unidad de lógica, se tiene comunicación a la API web para cargar puntajes, y con otros microcontroladores que se encargan solamente de administrar lo que debe mostrarse con los LEDs y las alertas sonoras.

Se puede visualizar el diseño general de la solución, a un nivel no tan técnico como el que se presentará más adelante, en el siguiente diagrama de despliegue. La aplicación, tanto back end como front y base de datos, está alojada en un servidor web; el front end se ejecuta en el navegador del dispositivo del usuario y le permite gestionar los datos necesarios y utilizar la AI. De manera separada, la máquina y su



control remoto se alojan físicamente en la Academia, se comunican entre sí por radio frecuencia y la máquina puede enviar datos la API del back end por medio de conexión Wi-Fi.

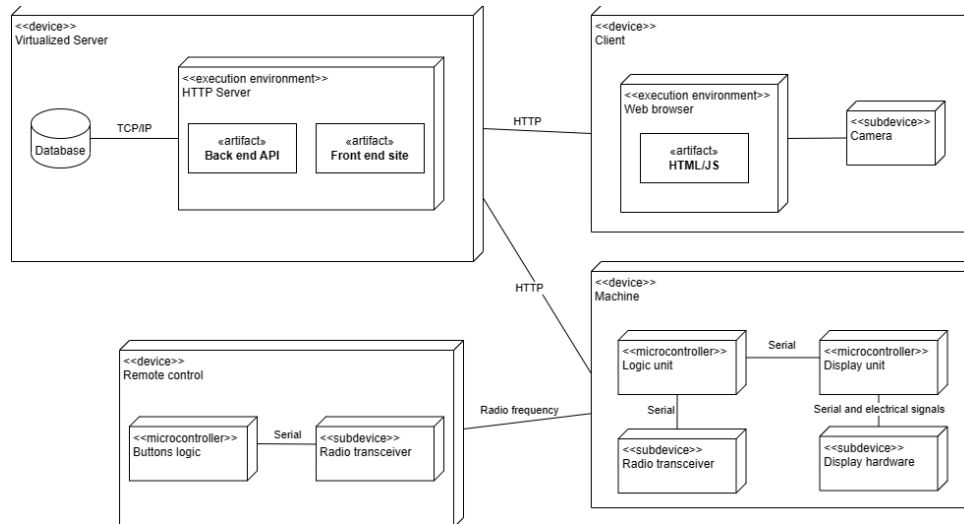


Diagrama 10 Diagrama de despliegue de la solución

7.1.3. Metodología de desarrollo

La metodología del proyecto está construida usando los principios de Agile para crear un manejo del proyecto que sea flexible y tolerante a cambios, y que a su vez permita la visibilidad de los avances para los Stakeholders.

El proyecto se está gestionando con un framework comúnmente conocido como ScrumBan, que es tomar del Scrum los sprints y el refinamiento de tareas, mientras se usa un Kanban Board para organizar todas las tareas que se deben hacer en el proyecto de principio a fin. Las ventajas de usar este framework es mantener la estructura del proyecto con Scrum, definiendo las metas que se deben ir entregando cada cierto tiempo, pero combinarlo con la flexibilidad y cambios frecuentes del Kanban, haciendo que los tickets que se generen en cada reunión con los Stakeholders puedan agregarse constantemente en el sprint actual o en cualquiera de los siguientes.

Debido a que el proyecto posee tres partes importantes que desarrollar, se decidió separar el liderazgo que tiene cada integrante a una sección del proyecto:

- Diego Hiriart tomaría liderazgo en lo que corresponde al módulo de inteligencia artificial y la máquina de conteo.
- Christian Samaniego se encargaría principalmente del diseño de los wireframes de la aplicación web y el desarrollo de los componentes en el front end.
- Luis Corales desarrollaría la parte correspondiente al back end de la aplicación web y el manejo de la base de datos.



De esta forma todos se ven involucrados en el desarrollo del proyecto, pues solo se tienen líderes de ciertas partes de la solución, pero todos aportan en todo desarrollo. Sin embargo, al asignar liderazgos, se permite que cada integrante pueda utilizar sus habilidades técnicas y aptitudes a un nivel mayor; trabajando en paralelo todas las partes del proyecto para avanzar uniformemente.

Como herramienta de control de versiones se utilizó Git que junto con Github permite trabajar colaborativamente y mantener un repositorio de código distribuido, además se lo utiliza en el presente proyecto para realizar control de cambios y correr acciones de integración continua.

Finalmente, como equipo se tiene en cuenta que la comunicación es clave para el desarrollo de los módulos, y es por ello que semanalmente se realizan dos reuniones con el tutor asignado, adicionales a las reuniones esporádicas que ocurran entre los miembros del proyecto para revisar tanto los avances, tareas nuevas y dudas con respecto a lo que se está desarrollando. Destacando que también existen canales donde está disponible la comunicación a todas horas, como mensajería de texto.

[Esta sección incluye el planteamiento de la solución, la metodología que se utilizará, La solución debe ser creativa e innovadora.]

7.2. Desarrollo de la solución

Tanto en las capas de presentación y de negocio se utilizó TypeScript, el cual es un lenguaje de programación que extiende la sintaxis de JavaScript para esencialmente agregar tipos estáticos. Para la implementación de la capa de presentación se utilizó React, una librería de JavaScript que facilita la creación de interfaces de usuario basadas en componentes. Facilita la generación de componentes dinámicos ya que utiliza un paradigma declarativo. Para la creación inicial del front-end se utilizó Vite, la cual permite construir de forma optimizada la aplicación de React junto con sus dependencias y adicionalmente correr servidores de desarrollo. La gestión de rutas de la aplicación desde el lado del cliente se implementó la librería React-router, la cual permite al usuario actualizar el URL desde el navegador y renderizar la página correcta sin realizar una petición al back end. Además, facilita la protección de rutas y control de acceso a través de roles. También se utiliza React Context API, la cual es una utilidad incluida en React, que permite manejar el estado global de la aplicación. Adicionalmente, se hace uso de Axios, el cual es un cliente HTTP que facilita el envío y recepción de peticiones hacia y desde el servidor. Para acelerar el tiempo de desarrollo se optó por utilizar Material UI, la cual es una librería de componentes y utilidades para React que facilitan la creación de interfaces de usuario.

A continuación, se detalla la estructura de la capa de presentación (front-end):

- **Components:** Comprende componentes reutilizables a través de múltiples páginas de la aplicación.



- **Contexts:** Incluye los componentes que hacen uso de ReactContext API los cuales son utilizados para que las variables de estado sean visibles por múltiples componentes y páginas.
- **Hooks:** Son funciones que brindan funcionalidad adicional a los componentes.
- **Pages:** Una página es el conjunto de componentes que integrados forman una vista de la aplicación.
- **Routes:** Son componentes que envuelven un conjunto de rutas relacionadas y pueden brindar validaciones o funcionalidades adicionales.
- **Services:** Consta directorios de utilidades varias. Aquí se incluye la configuración del cliente HTTP Axios.

En paralelo, se fue desarrollando el back end para la obtención y registro de datos en la base de datos. Para ello se inicializo Prisma, una librería ORM (Object Relational Mapping) que permitió la creación de la base de datos en PostgreSQL 15. A continuación se mostrará el diagrama Entidad-Relación del proyecto:

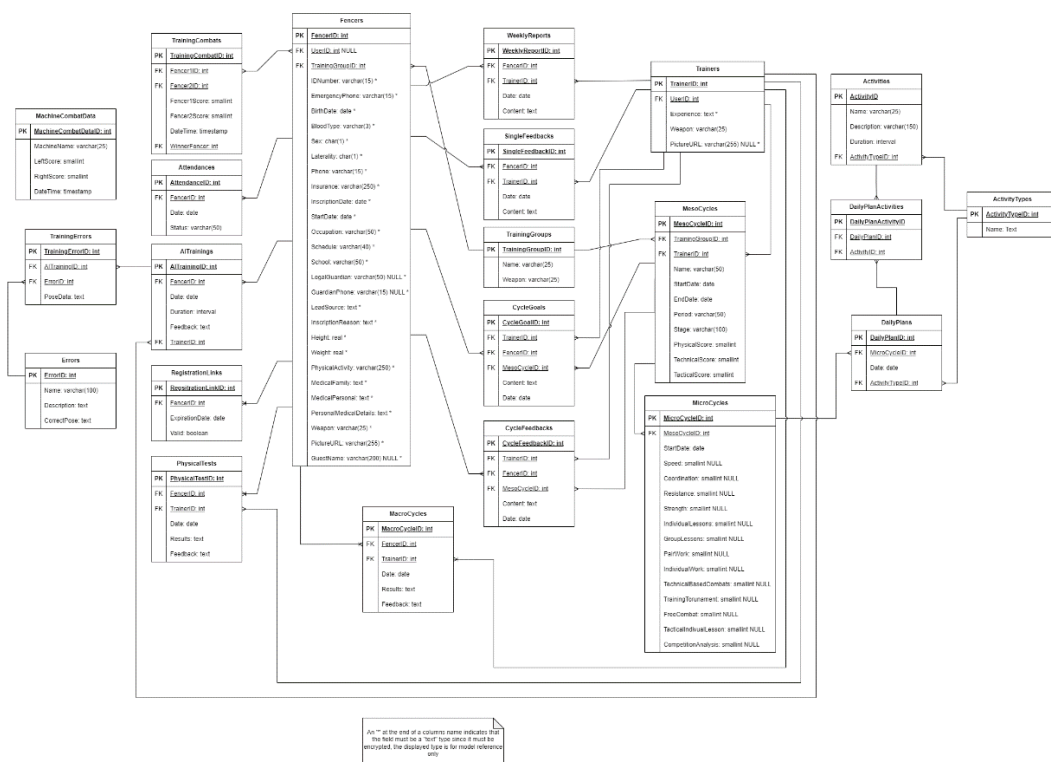


Diagrama 11 Diagrama Entidad-Relación de la solución

Para hacer la conexión entre el back end y el front end se desarrolló una Restful API con el framework web Express para Typescript. La idea detrás de esta implementación es tener una API abierta a peticiones que provengan de distintas aplicaciones y sistemas que el proyecto posee, siendo el front end, la maquina y el módulo de inteligencia artificial que trabajan por separado, pero al final de cumplir sus funciones todas deben apuntar al back end que sirve para la consulta y guardado de datos.



Teniendo en cuenta que esta parte del proyecto fue desarrollada con un diseño por capas, tenemos la siguiente estructura:

- **Routes:** se reciben las peticiones a distintos endpoints que permiten ejecutar alguna función, relacionada a la base de datos o no. Estas rutas están divididas en tres grupos: rutas abiertas (pueden ser consultadas sin estar autenticado en el sistema), rutas de autenticación y rutas protegidas (todos los endpoints que necesitan un token de autenticación para tener alguna respuesta).
- **Middlewares:** estos son funciones que se ejecutan en el momento que se recibe la petición en algún endpoint y antes de que se dé una respuesta al usuario. En esta capa se definen la distinción de roles que hay en los usuarios, la lógica que permite a ciertos usuarios ser admitidos como administrador, esgrimista o entrenador. También se define el middleware que verifica el token de autenticación que el usuario debe tener para ciertas rutas.
- **Controllers:** esta capa es el orquestador de funciones que el endpoint posee, es decir, la lógica detrás de cada ruta definida. Los controladores son ejecutados cuando la petición que llegó ha pasado exitosamente por todos los middlewares definidos en la ruta. Esta capa ejecuta funciones necesarias para la aplicación funciones CRUD de una tabla de la base de datos.
- **Data:** es la capa donde se definen todos los CRUDs de la base de datos. Esta capa es llamada por los Controllers según la necesidad de la ruta.
- **Utils:** funciones que realizan algún proceso específico que puede ser reutilizado en cualquier otra capa. Por ejemplo: el hashing de las contraseñas, la creación de un token JWT (JSON Web Token), el logging de errores en el servidor y formatear fechas.

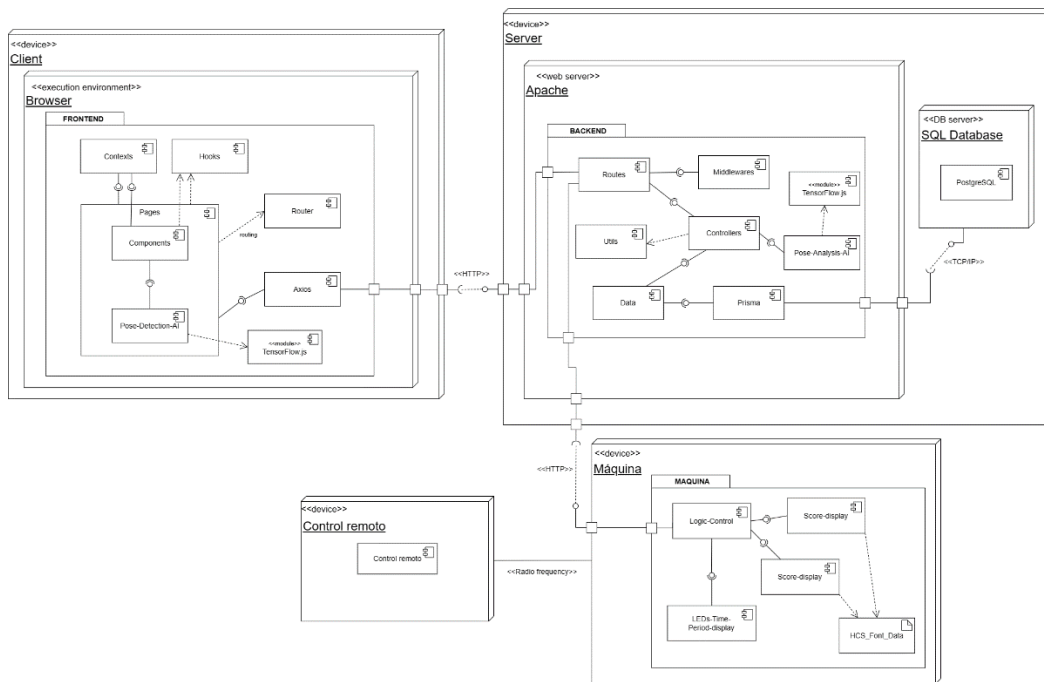


Diagrama 12 Diagrama de componentes

Se implementaron tres roles principales que serán vinculados a los usuarios de la aplicación:

- Esgrimista: Serán vinculados a los alumnos de la academia. Se les asigna una planificación y son capaces de observar retroalimentación de sus entrenadores, participar en combates y realizar entrenamientos asistidos por IA.
- Entrenador: Serán vinculados a los entrenadores de la academia. Serán capaces de gestionar planificaciones, grupos, esgrimistas, llevar a cabo combates y generar retroalimentación.
- Administrador: Son los responsables de la gestión de los usuarios tanto de entrenadores como de esgrimistas. Además, comparte funciones con el entrenador.

Con el fin de optimizar el flujo de trabajo de los entrenadores se optó por implementar un proceso de registro de usuarios a través de un link de invitación. Este modo de registro permite a un entrenador generar un link de registro con una determinada fecha de expiración y de un solo uso. Este link es usado por los esgrimistas que deseen registrarse en la academia y utilizar la aplicación.

La inteligencia artificial de detección de posibles errores, implementada con la librería TensorFlow.js, está integrada tanto en el front end (y por ende el dispositivo del cliente), utiliza un modelo de AI pre entrenado por TensorFlow que se especializa en detectar la pose humana en una imagen. Usando la cámara del dispositivo donde se ejecute la aplicación web, se realiza el análisis de varias poses por segundo. Las poses



extraídas contienen datos de 33 puntos en el cuerpo humano con coordenadas x, y, z que indican las posiciones de cada punto con respecto al centro de la cadera como origen. Estas poses tridimensionales, al unirse, indican la secuencia de pasos que realizó un esgrimista en un periodo de tiempo de entre uno a dos segundos. Una vez agrupadas suficientes de estas poses, se envían al back end para ser analizados por un modelo personalizado de inteligencia artificial. Cuando la respuesta se obtiene, si se detectó un error, se pausa el entrenamiento para mostrar el error cometido y cómo es el movimiento correcto; caso contrario se continua el entrenamiento

El análisis de las poses es realizado por un modelo creado específicamente para este propósito, usando TensorFlow para generar una red neuronal recurrente. La red utiliza “long short-term memory” (LSTM) que le permite aprender dependencias entre datos, en este caso poses. De esta manera, la red puede considerar las poses ya realizadas por el esgrimista para determinar si el movimiento es erróneo. Siendo un modelo de clasificación, cuando el front end envía el conjunto de poses, la inteligencia artificial del back end determina si se cometió un error y cual fue. En el caso de detectarse un error, se guarda el mismo en la base y se lo retorna junto con el movimiento correcto.

[Esta sección incluye la aplicación del diseño de ingeniería para el desarrollo del prototipo, aplicación, producto etc. Incluir aplicación de buenas prácticas, estándares, códigos de ingeniería, restricciones de diseño, entre otros.]

7.3. Pruebas y evaluación de la solución

7.3.1. Validaciones con usuarios

7.3.2. Continuous integration

El proyecto consta con un archivo de configuración correspondiente al pipeline de CI (Continuous integration). Este pipeline ejecuta un workflow que verifica el linter implementado en la aplicación web (corregir el formato del código, errores de sintaxis, etc), la ejecución de las pruebas unitarias y la compilación del proyecto completo antes de poder hacer merge en la rama principal del repositorio.

[Esta sección incluye las pruebas a las que se sometió la solución, los resultados obtenidos y las mejoras en cada iteración. Debe ser un proceso iterativo y de mejora continua.]

7.4. Resultados y Discusión.

[Esta sección incluye las conclusiones que se derivan de los objetivos planteados, además, las recomendaciones que se pueden desprender al final de la ejecución del proyecto.]



7.5. Implicaciones éticas

[Esta sección incluye los temas éticos que se consideran en la solución y las implicaciones a las que se puede enfrentar el desarrollo y ejecución del proyecto.]

8. Conclusiones y Recomendaciones

[Esta sección incluye las conclusiones que se derivan de los objetivos planteados, además, las recomendaciones que se pueden desprender al final de la ejecución del proyecto.]

9. Trabajo futuro

[Esta sección incluye los posibles proyectos que se pueden generar a partir de los resultados de este.]

10. Referencias bibliográficas

Absolute Fencing Gear. (2022). *St. George 11*. Obtenido de Absolute Fencing Gear: <https://www.absolutefencinggear.com/st-george-11.html>

Absolute Fencing Gear. (2022). *St. George 12 Plus*. Obtenido de Absolute Fencing Gear: <https://www.absolutefencinggear.com/st-george-12-plus.html>

Athlete Analyzer. (2022). *About us / Athlete Analyzer*. Obtenido de Athlete Analyzer: <https://www.athleteanalyzer.com/about-us>

CoachUp. (2023). Retrieved from CoachUp: <https://www.coachup.com/>

FIE (2022). *Men's Individual Epee - Tokyo 2020 Olympics*. Obtenido de FIE International Fencing Federation: https://static.fie.org/uploads/25/127828-219585984_4213960165318462_3571882008116433051_n.jpg

FIFA. (08 de Diciembre de 2022). *Video Assistant Referee (VAR)*. Obtenido de FIFA: <https://www.fifa.com/technical/football-technology/football-technologies-and-innovations-at-the-fifa-world-cup-2022/video-assistant-referee-var>

FlexIt Fitness. (2023). *FlexIt*. Retrieved from Interconnected Health: <https://flexit.fit/virtualpt/>

G2. (2023). *Best Athlete Management Software*. Obtenido de G2: <https://www.g2.com/categories/athlete-management>

K. Apostolou, C. T. (2019). Sports Analytics algorithms for performance prediction. *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)* (págs. 1-4). Patras, Greece: IEEE.

Liu, Q. (2022). Aerobics posture recognition based on neural network and sensors. *Neural Computing and Applications*, 34(5), 3337–3348. doi:<https://doi.org/10.1007/s00521-020-05632-w>



- Muñoz, B. (2021). *Desarrollo y validación de un sistema sin marcadores para el análisis del movimiento humano*. Obtenido de <https://riunet.upv.es/handle/10251/173478>
- Olympia Fencing Center. (2021). *All About Fencing*. Obtenido de Olympia Fencing Center: <https://olympiafencingcenter.com/discover-the-olympic-sport-of-fencing/>
- V. Sarlis, C. T. (2020). Sports analytics — Evaluation of basketball players and team performance. *Information Systems*, 93, 1-30.
doi:<https://doi.org/10.1016/j.is.2020.101562>
- Zenia. (2022). *Zenia app*. Retrieved from <https://zenia.app>

11. Anexos

11.1. Anexo 1: documentos de la FIE

<https://fie.org/fie/documents/statutes-and-admin-rules>

11.2. Anexo 2: sitio web de la Academia de Esgrima Ciudad de Quito

<https://quitoesgrima.com/>

11.3. Anexo 3: identificación de tareas de proyecto y estimaciones

Para cada paquete de trabajo, es decir pro cada entregable, se han identificado actividades necesarias para el desarrollo de cada paquete. Estas actividades se identifican y describen a continuación, junto con la estimación por tres valores para cada una.

1. Requerimientos

1.1. Documentación de requerimientos

1.1.1. Recopilación de requerimientos

Obtención y especificación de requerimientos de software que estén alineados con las necesidades actuales de la Academia.

Estimación optimista: 1 semana, estimación más probable: 2 semanas, estimación pesimista: 3 semanas. Media: 2.

1.1.2. Generar documentos preliminares y realizar correcciones

Comprende actividades de análisis, refinamiento y validación de requisitos.

Estimación optimista: 1 semana, estimación más probable: 2 semanas, estimación pesimista: 3 semanas. Media: 2.

1.1.3. Crear documentación final de requerimientos

Documentar los requerimientos definitivos del proyecto.



Estimación optimista: 1 semana, estimación más probable: 2 semanas, estimación pesimista: 3 semanas. Media: 2.

2. Aplicación web

2.1. API de la aplicación web (backend)

2.1.1. Selección de arquitectura

Determinar la combinación de herramientas, aplicaciones y servicios que serán utilizados durante el desarrollo de la aplicación web.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 2 semanas. Media: 1.12

2.1.2. Diseño de software

Definir la estructura del sistema y la relación entre sus elementos haciendo uso de patrones de diseño y buenas prácticas de desarrollo.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 2 semanas, Media: 1.12

2.1.3. Implementación de API

Desarrollar y exponer las funcionalidades de la aplicación a través de un API que podrá ser consumida por los demás módulos del sistema.

Estimación optimista: 2 semanas, estimación más probable: 3 semanas, estimación pesimista: 4 semanas. Media: 3

2.2. Base de datos

2.2.1. Elaboración de diagrama entidad relación

Ilustrar las relaciones entre las entidades identificadas en la base de datos.

Estimación optimista: 3 días, estimación más probable: 5 días, estimación pesimista: 1 semana. Media: 0.71

2.2.2. Normalización de base de datos

Emplear reglas de normalización para asegurar el buen diseño y mantenibilidad de la base de datos.

Estimación optimista: 3 días, estimación más probable: 5 días, estimación pesimista: 1 semana. Media: 0.71

2.2.3. Implementación de base de datos

Selección de un motor de base de datos, creación de tablas y relaciones necesarias para cumplir con la lógica del negocio.

Estimación optimista: 3 días, estimación más probable: 5 días, estimación pesimista: 1 semana. Media: 0.71



2.3. Interfaz de página web (frontend)

2.3.1. Diseño de estructura/wireframes

Crear una representación visual de la estructura y funcionalidad inicial del sistema.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 1.5 semanas. Media: 1.06

2.3.2. Diseño de estilos

Determinar el estilo de los componentes que serán usados en la aplicación web que aseguren usabilidad y accesibilidad.

Estimación optimista: 3 días, estimación más probable: 5 días, estimación pesimista: 1 semana. Media: 0.71

2.3.3. Implementación de página web

Desarrollar las interfaces y componentes interactivos que consumirán los servicios del backend (API).

Estimación optimista: 2 semanas, estimación más probable: 3 semanas, estimación pesimista: 5 semanas. Media: 3

3. Detección de posibles errores

3.2. Dataset de entrenamiento para la AI

3.2.1. Capturar fotos y video

Obtener material fotográfico y en video de errores en acciones simples de esgrima que serán utilizadas durante el proceso de entrenamiento.

Estimación optimista: 3 días, estimación más probable: 5 días, estimación pesimista: 1 semana. Media: 0.71. = 5d

3.2.2. Identificar y marcar errores en los datos

Marcar los movimientos erróneos de interés dentro de las imágenes recopiladas que correspondan a errores comunes de esgrima. Las imágenes que se usarán son las fotos obtenidas anteriormente y cuadros específicos de los videos donde se evidencian los errores. A partir de estos errores preidentificados, la inteligencia artificial puede ser entrenada.

Estimación optimista: 3 días, estimación más probable: 5 días, estimación pesimista: 1 semanas. Media: 0.71



3.3. Modelo de AI para reconocimiento de posibles errores

3.3.1. Diseñar modelo

Determinar las herramientas y algoritmos que serán utilizados en el modelo de aprendizaje.

Estimación optimista: 1 semana, estimación más probable: 1.5 semanas, estimación pesimista: 3 semanas. Media: 1.66

3.3.2. Implementar modelo

Desarrollo del modelo de reconocimiento de posibles errores e integración de este con la aplicación web.

Estimación optimista: 1 semana, estimación más probable: 2 semanas, estimación pesimista: 3.5 semanas. Media: 2.08

3.3.3. Entrenar modelo

Alimentar el conjunto de datos de entrenamiento al modelo para que se realice el aprendizaje necesario para la identificación.

Estimación optimista: 3 día, estimación más probable: 1 semana, estimación pesimista: 2 semanas. Media: 1.07

3.3.4. Almacenar modelo para utilización

Almacenar los pesos y parámetros de las capas de la red neuronal para que puedan ser reutilizados siempre que se requiera hacer detección de probables errores.

Estimación optimista: 3 días, estimación más probable: 1 semana, estimación pesimista: 2 semanas. Media: 1.07

3.4. AI de reconocimiento integrada en la app web

3.4.1. Diseñar software para envío de datos

Determinar el diseño que utilizará el software encargado de enviar datos de posibles errores detectados a la aplicación web usando los resultados de reconocimiento de la inteligencia artificial.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 1.5 semanas. Media: 1.04

3.4.2. Implementar software de envío

Desarrollar el software encargado de la comunicación con la aplicación web.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 1.5 semanas. Media: 1.04



4. Conteo de puntajes

4.1. Hardware de máquinas

4.1.1. Determinar módulos requeridos

Identificar los componentes de hardware necesarios para la construcción de las máquinas.

Estimación optimista: 1 día, estimación más probable: 2 días, estimación pesimista: 3 días. Media: 0.29

4.1.2. Adquisición de partes

Se deben adquirir las partes para el hardware, de acuerdo con los módulos identificados. Dependiendo de costos y disponibilidad, algunas partes se podrán obtener localmente, otras podrían tener que ser importadas.

Estimación optimista: 1 semana, estimación más probable: 2.5 semanas, estimación pesimista: 3.5 semanas. Media: 2.33 semanas

4.1.3. Crear planos y diagramas de conexión

Crear diagramas que ilustren la disposición de los componentes de hardware y sus conexiones.

Estimación optimista: 1 día, estimación más probable: 3 días, estimación pesimista: 5 días. Media: 0.49

4.1.4. Ensamblar hardware

Conexión y ensamblaje de los componentes de hardware, incluye cargar el software compilado al microcontrolador.

Estimación optimista: 1 semana, estimación más probable: 1.5 semanas, estimación pesimista: 2 semanas. Media: 1.5

4.2. Software de máquinas

4.2.1. Diseñar software

Determinar el diseño del software que será utilizado por el microcontrolador para el uso componentes de hardware en las máquinas.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 2 semanas. Media: 1.12

4.2.2. Identificar librerías de control

Identificar librerías necesarias que habiliten la utilización y control de los componentes de hardware por parte del microcontrolador.

Estimación optimista: 1 día, estimación más probable: 3 días, estimación pesimista: 5 días. Media: 0.43



4.2.3. Implementar software de conteo

Desarrollar el firmware que será utilizado para controlar los componentes de hardware, mantener conteo de los puntos, e input del usuario.

Estimación optimista: 1 semana, estimación más probable: 2 semana, estimación pesimista: 3 semanas. Media: 2

4.3. Funcionalidad de registro de puntaje en aplicación web

4.3.1. Diseño de endpoint (API) y frontend

Determinar el diseño para el endpoint necesario en la API (backend), y el diseño de la interfaz de usuario.

Estimación optimista: 5 días, estimación más probable: 1 semanas, estimación pesimista: 2 semanas. Media: 1.12

4.3.2. Implementar endpoint de función

Codificar el endpoint para el guardado de resultados en la base de datos del sistema.

Estimación optimista: 5 días, estimación más probable: 1 semana, estimación pesimista: 2 semanas. Media: 1.12

4.3.3. Implementar frontend de función

Añadir, a la aplicación web del sistema, una funcionalidad (menú, opción) para registrar los resultados de los combates que se tengan en las máquinas.

Estimación optimista: 5 días, estimación más probable: 1 semanas, estimación pesimista: 2 semanas. Media: 1.12

11.4. Anexo 4: planificación en Microsoft Project

Documento de planificación: [Capstone Project planning Hiriart Corales Samaniego.mpp](#)