



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

16 de julio de 2023

Quito, Ecuador



Índice

RESUMEN	IV
ABSTRACT	V
1. INTRODUCCIÓN	1
1.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA O NECESIDAD	1
1.2. DESCRIPCIÓN DE LA EMPRESA	3
1.3. IMPACTO DEL PROYECTO EN LA SOCIEDAD	4
2. ANÁLISIS DE POSIBLES SOLUCIONES	5
2.1. IDENTIFICACIÓN Y SELECCIÓN DE LA MEJOR SOLUCIÓN	5
3. OBJETIVOS	7
3.1. OBJETIVO GENERAL	7
3.2. OBJETIVOS ESPECÍFICOS	7
4. ALCANCE	7
4.1. ALCANCE DE LA SOLUCIÓN SELECCIONADA	7
4.2. LIMITACIONES Y RESTRICCIONES DEL PROYECTO	10
4.2.1. Limitaciones	10
4.2.2. Supuestos	11
4.2.3. Restricciones	11
5. PLANIFICACIÓN Y COSTOS DEL PROYECTO	12
5.1. ESTRUCTURA DE DESGLOSE DE TRABAJO (EDT)	12
5.2. DESCRIPCIÓN DE PAQUETES DE TRABAJO	12
1. Requerimientos	12
1.1. Documentación de requerimientos	12
2. Aplicación web	13
2.1. API de la aplicación web (backend)	13
2.2. Base de datos	13
2.3. Interfaz de página web (frontend)	13
3. Detección de posibles errores	13
3.1. Dataset de entrenamiento para la AI	13
3.2. Modelo de AI para reconocimiento de posibles errores	13
3.3. AI de reconocimiento integrada en la app web	13
4. Conteo de puntajes	13
4.1. Hardware	13
4.2. Software de máquinas	13
4.3. Funcionalidad de registro de puntaje en aplicación web	13
5.3. DIAGRAMA DE GANTT	14
5.4. ANÁLISIS DE COSTOS DE LA SOLUCIÓN	15
6. DESCRIPCIÓN DE ESTUDIOS REALIZADOS	17
7. DESARROLLO DE LA SOLUCIÓN	18
7.1. DISEÑO DE LA SOLUCIÓN	18
7.1.1. Contexto	18
7.1.2. Requerimientos	19
7.1.3. Diseño y arquitectura	28
A. Aplicación web	28



B.	Inteligencia artificial para detección de movimientos correctos	30
C.	Máquina de puntaje y temporizador	32
D.	Vista general de la solución	33
7.1.4.	<i>Metodología de desarrollo</i>	34
7.2.	DESARROLLO DE LA SOLUCIÓN	41
7.2.1.	<i>Ambiente de desarrollo</i>	41
7.2.2.	<i>Implementación</i>	42
A.	Aplicación web	42
B.	Inteligencia artificial para entrenamiento	51
C.	Máquina de puntaje y temporizador	55
7.2.3.	<i>Despliegue</i>	60
7.3.	PRUEBAS Y EVALUACIÓN DE LA SOLUCIÓN	61
7.3.1.	<i>Proceso de pruebas continuas</i>	61
7.3.2.	<i>Validaciones con usuarios</i>	61
A.	Diseño	62
1)	Aplicación web	62
2)	Máquina de puntaje	62
B.	Flujo de funcionalidades	62
1)	Aplicación web	62
2)	Máquina de puntaje	63
C.	Pruebas de aceptación	63
1)	Aplicación web	63
2)	Máquina de puntaje	63
7.3.3.	<i>Integración continua</i>	63
7.4.	RESULTADOS Y DISCUSIÓN	64
7.5.	IMPLICACIONES ÉTICAS	67
8.	CONCLUSIONES Y RECOMENDACIONES	67
8.1.	CONCLUSIONES	67
8.2.	RECOMENDACIONES	68
9.	TRABAJO FUTURO	68
10.	REFERENCIAS BIBLIOGRÁFICAS	70
11.	ANEXOS	71
11.1.	ANEXO 1: DOCUMENTOS DE LA FIE	71
11.2.	ANEXO 2: SITIO WEB DE LA ACADEMIA DE ESGRIMA CIUDAD DE QUITO	71
11.3.	ANEXO 3: IDENTIFICACIÓN DE TAREAS DE PROYECTO Y ESTIMACIONES	71
1.	<i>Requerimientos</i>	72
1.1.	Documentación de requerimientos	72
1.1.1.	Recopilación de requerimientos	72
1.1.2.	Generar documentos preliminares y realizar correcciones	72
1.1.3.	Crear documentación final de requerimientos	72
2.	<i>Aplicación web</i>	72
2.1.	API de la aplicación web (backend)	72
2.1.1.	Selección de arquitectura	72
2.1.2.	Diseño de software	72
2.1.3.	Implementación de API	72
2.2.	Base de datos	73
2.2.1.	Elaboración de diagrama entidad relación	73
2.2.2.	Normalización de base de datos	73
2.2.3.	Implementación de base de datos	73
2.3.	Interfaz de página web (frontend)	73
2.3.1.	Diseño de estructura/wireframes	73



2.3.2.	Diseño de estilos	73
2.3.3.	Implementación de página web	73
3.	<i>Detección de posibles errores</i>	74
3.1.	Dataset de entrenamiento para la AI	74
3.1.1.	Capturar fotos y video	74
3.1.2.	Identificar y marcar errores en los datos	74
3.2.	Modelo de AI para reconocimiento de posibles errores	74
3.2.1.	Diseñar modelo	74
3.2.2.	Implementar modelo	74
3.2.3.	Entrenar modelo	74
3.2.4.	Almacenar modelo para utilización	74
3.3.	AI de reconocimiento integrada en la app web	75
3.3.1.	Diseñar software para envío de datos	75
3.3.2.	Implementar software de envío	75
4.	<i>Conteo de puntajes</i>	75
4.1.	Hardware de máquinas	75
4.1.1.	Determinar módulos requeridos	75
4.1.2.	Adquisición de partes	75
4.1.3.	Crear planos y diagramas de conexión	75
4.1.4.	Ensamblar hardware	75
4.2.	Software de máquinas	76
4.2.1.	Diseñar software	76
4.2.2.	Identificar librerías de control	76
4.2.3.	Implementar software de conteo	76
4.3.	Funcionalidad de registro de puntaje en aplicación web	76
4.3.1.	Diseño de endpoint (API) y frontend	76
4.3.2.	Implementar endpoint de función	76
4.3.3.	Implementar frontend de función	76
11.4.	ANEXO 4: PLANIFICACIÓN EN MICROSOFT PROJECT	77
11.5.	ANEXO 5: ESQUEMÁTICA DE CONEXIÓN DE LA MÁQUINA	78
11.5.1.	<i>Unidad de lógica y displays</i>	78
11.5.2.	<i>Control remoto</i>	83



Resumen

La Academia de Esgrima Ciudad de Quito es una institución dedicada a la formación de esgrimistas. Actualmente manejan todos los datos de sus deportistas (planes de entrenamiento, desempeño, y datos personales) de manera manual o en ocasiones sin registro alguno. Por otra parte, la limitada cantidad de entrenadores disponibles reduce la posibilidad de que los esgrimistas reciban retroalimentación oportuna durante sus entrenamientos individuales, dado que no siempre cuentan con una manera de registrar sus prácticas o ser supervisados. Además, no se cuenta con una manera confiable de registrar tiempo o puntajes durante los combates a excepción de la memoria del árbitro presente. Considerando esto, el presente proyecto tiene como objetivo desarrollar una aplicación web utilizando una arquitectura por capas para la Academia, la cual permita la gestión de datos de entrenamiento. Esta aplicación web integra una inteligencia artificial para detectar potenciales errores durante entrenamientos individuales. Adicionalmente, se desarrolla una máquina de puntajes utilizando microcontroladores, displays e indicadores LED para la gestión de puntajes, tiempo y penalizaciones durante los combates de entrenamiento. La implementación de la aplicación web en la academia ha permitido la digitalización de los datos y optimización de sus procesos, ofreciendo una mejor gestión de entrenamientos y acceso a su información. Por otro lado, la inteligencia artificial ha permitido a los esgrimistas obtener una retroalimentación más conveniente durante sus sesiones de entrenamiento. De igual manera, los entrenadores tienen mejores opciones de monitoreo de desempeño de sus alumnos. Finalmente, la máquina de puntajes desarrollada ha permitido brindar una alternativa de costo más accesible que facilita el flujo de gestión de puntos, penalización, y tiempo durante combates.

Palabras clave: deporte, esgrima, aplicación web, JavaScript, TypeScript, Express.js, React, inteligencia artificial, TensorFlow.js, microcontrolador, Arduino, ESP32, electrónica



Abstract

The “Academia de Esgrima Ciudad de Quito” (Ciudad de Quito Fencing Academy) is an institution dedicated to the training of fencers. Currently they manage all the data of their athletes (training plans, performance, and personal data) manually or sometimes without any record. On the other hand, the limited number of coaches available reduces the possibility for fencers to receive timely feedback during their individual training, since they do not always have a way to record their practices or be supervised. Furthermore, there is no reliable way to keep time or scores during bouts except for the memory of the present referee. Considering this, this project aims to develop a web application using a layered architecture for the Academy, which allows the management of training data. This web application integrates artificial intelligence to detect potential errors during individual training sessions. Additionally, a scoring machine is developed using microcontrollers, displays and LED indicators for scoring, time and penalty management during training matches. The implementation of the web application in the academy has allowed the digitization of data and optimization of its processes, offering better training management and access to its information. On the other hand, artificial intelligence has allowed fencers to get more convenient feedback during their training sessions. Similarly, coaches have better performance monitoring options for their students. Finally, the developed scoring machine has made it possible to provide a more affordable alternative that facilitates the management flow of points, penalties, and time during combats.

Key words: sports, fencing, web application, JavaScript, TypeScript, Express.js, React, artificial intelligence, TensorFlow.js, microcontroller, Arduino, ESP32, electronics



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)

uol/a.



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

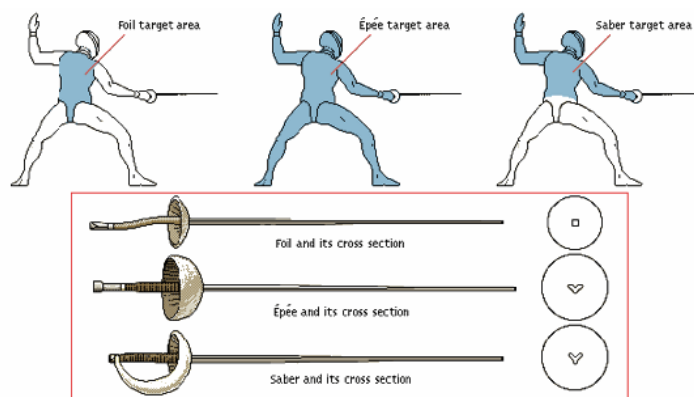


Imagen 3 Diferencias entre las 3 armas de esgrima y los blancos válidos para cada una (Olympia Fencing Center, 2021)

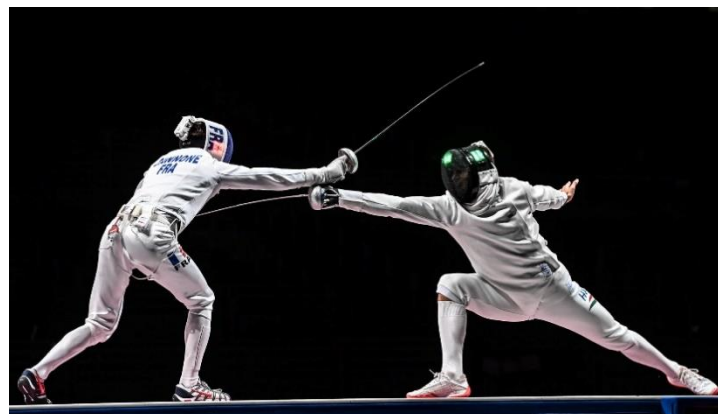


Imagen 4 Combate de espadistas en las Olimpiadas de Tokio 2020 (FIE, 2022a)

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 mejoras 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:

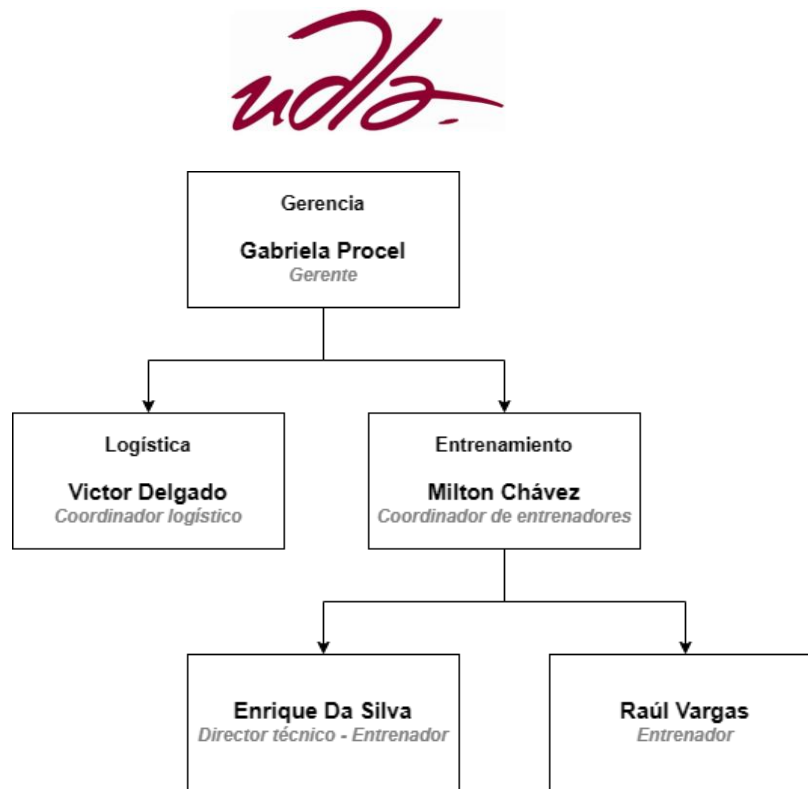


Diagrama 1 Organizagrama de la Academia de Esgrima Ciudad de Quito

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:
 - 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

4.2.1. 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.

4.2.2. 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.

4.2.3. 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).

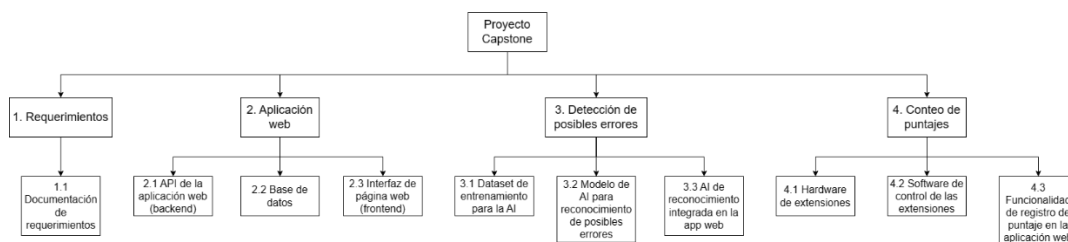


Imagen 5 Diagrama de estructura de desglose de trabajo (EDT)

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](#).

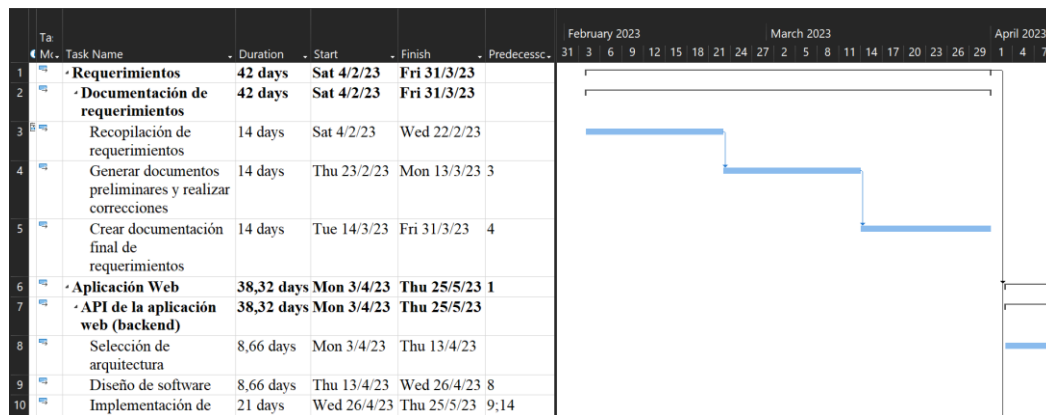


Diagrama 2 Planificación en Diagrama de Gantt 1/6

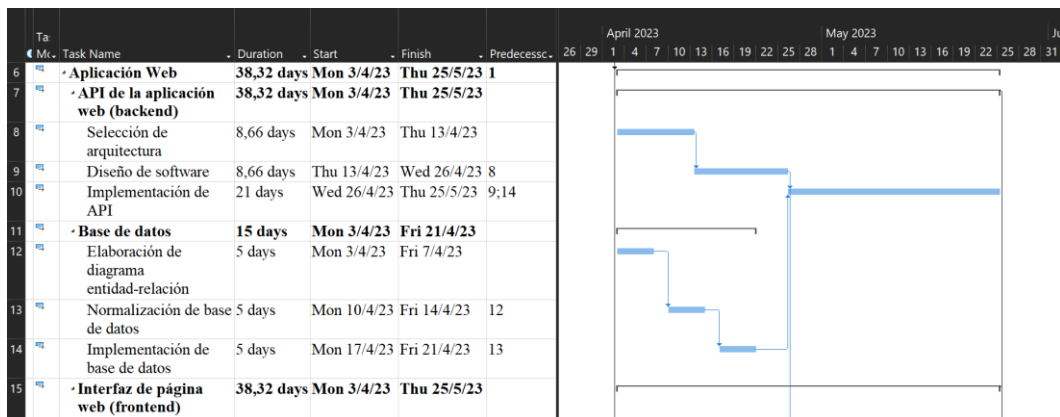


Diagrama 3 Planificación en Diagrama de Gantt 2/6

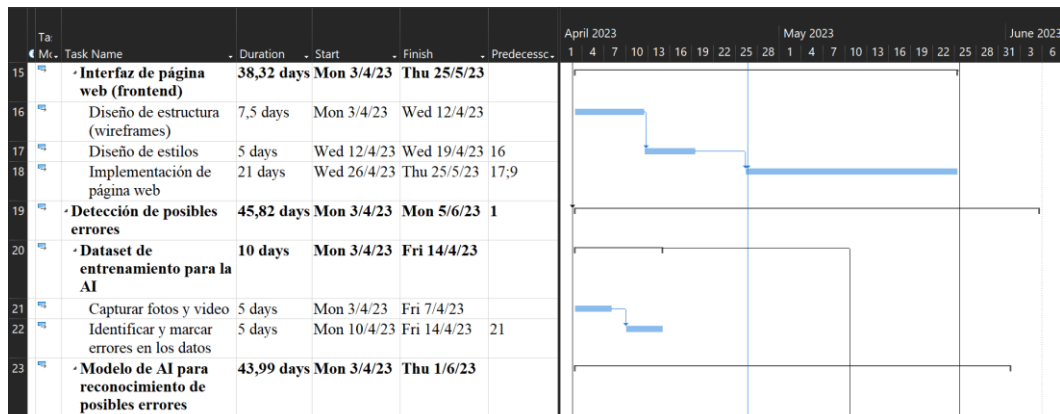


Diagrama 4 Planificación en Diagrama de Gantt 3/6

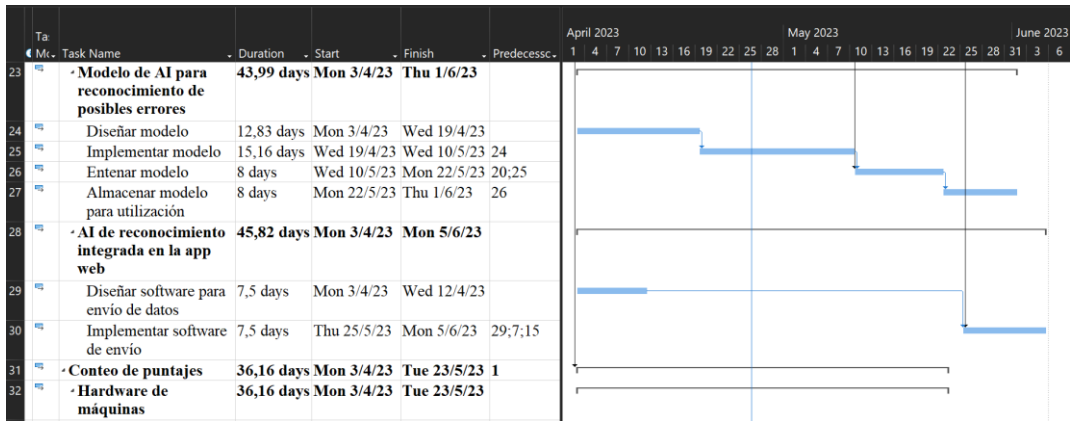


Diagrama 5 Planificación en Diagrama de Gantt 4/6

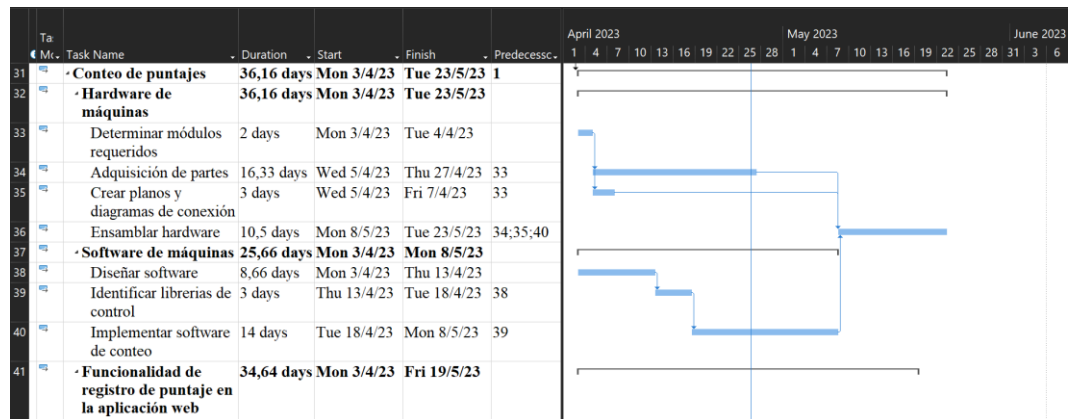


Diagrama 6 Planificación en Diagrama de Gantt 5/6

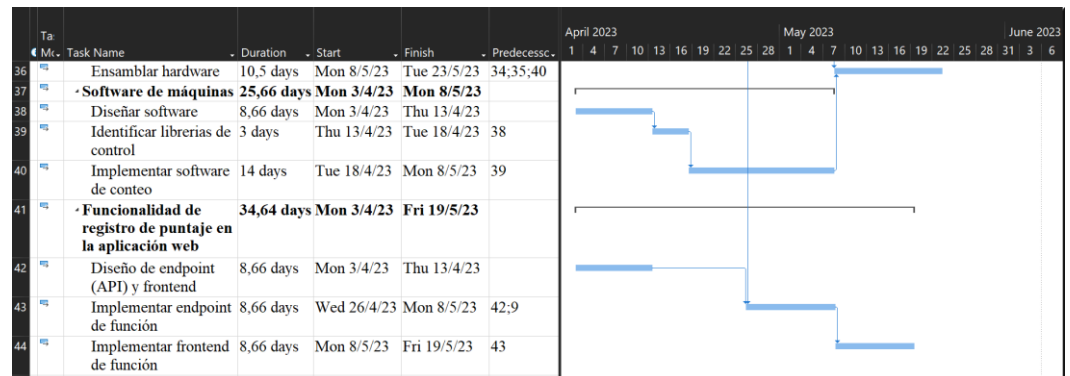


Diagrama 7 Planificación en Diagrama de Gantt 6/6

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. Contexto

La solución tiene, esencialmente, dos componentes que funcionan de manera separada. Primero, la aplicación web que permite la gestión de entrenamientos y datos de los esgrimistas y su desempeño; la misma que incluye la funcionalidad para que un esgrimista lleve a cabo entrenamientos con inteligencia artificial. La aplicación web es utilizada por 3 tipos de usuario:

- El administrador: Puede gestionar esgrimistas y entrenadores, así como como sus datos.
- El entrenador: Puede gestionar esgrimistas, planes de entrenamiento, objetivos y retroalimentaciones.
- El esgrimista: Puede gestionar sus propios datos, consultar planes, y usar el entrenamiento con inteligencia artificial.

Cómo segundo elemento está la máquina de puntaje y temporizador. Esta máquina permite controlar el tiempo transcurrido en combates, editar los puntajes de cada esgrimista, asignar penalizaciones (con el efecto en los puntajes que estas tienen), y asignar prioridad de manera aleatoria (una función para resolver empates, siguiendo reglamentos de esgrima). Todo esto se puede controlar con el uso de un control remoto, mismo que usaría el entrenador (o arbitro en el escenario de un combate) para utilizar las funcionalidades antes descritas.

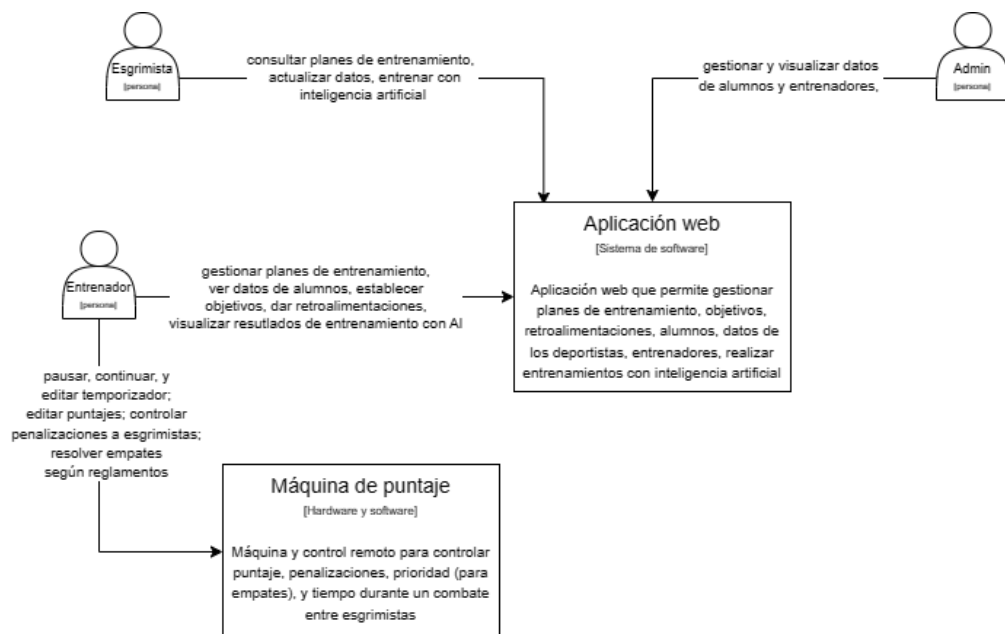


Diagrama 8 Diagrama de contexto de la solución



7.1.2. 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:

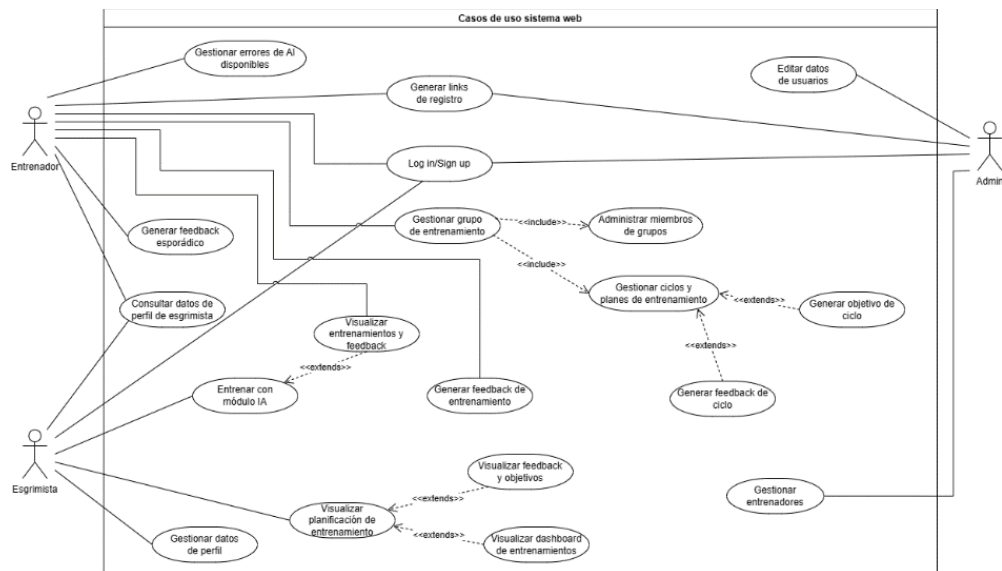


Diagrama 9 Casos de uso para app web

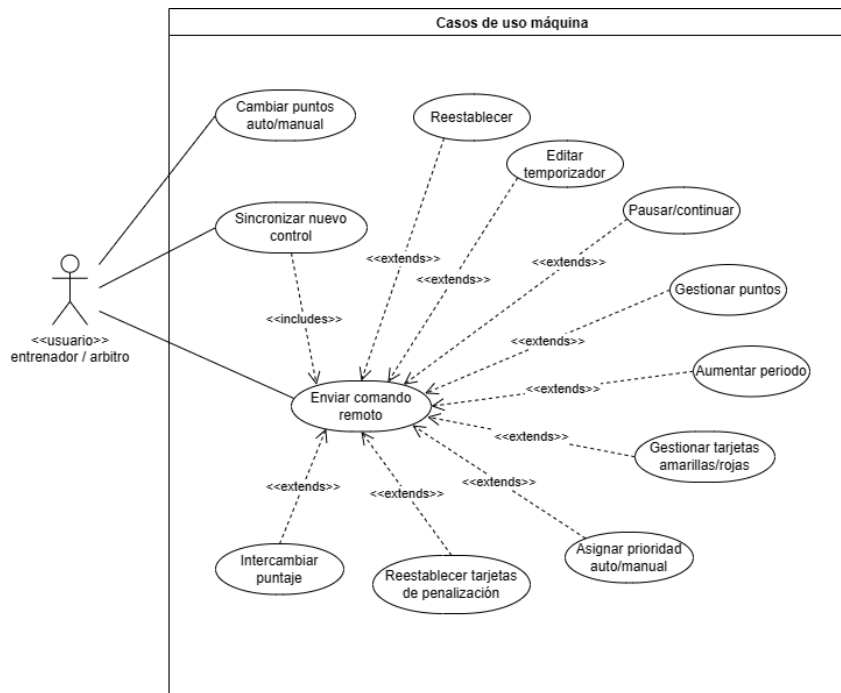


Diagrama 10 Casos de uso para máquina de conteo

A continuación, se listan los casos de uso del proyecto en tablas que contienen el nombre del caso de uso, un identificador, su descripción, la meta y los actores que participan en ese caso:

Nombre Caso de Uso:	Log in/Sign up
Identificador:	CS-01
Descripción:	El sistema debe permitir que los usuarios se registren y autenticuen en la aplicación web.
Meta:	Permite a los usuarios ingresar a la aplicación usando sus datos.
Actor:	Administrador, entrenador, esgrimista

Tabla 4 Caso de uso app web 1 Log in / Sign up

Nombre Caso de Uso:	Generar links de registro
Identificador:	CS-02
Descripción:	El sistema debe permitir la generación de links de registro a usuarios que están ingresando a la academia.
Meta:	Permite que los administradores y entrenadores generen un link a nuevos usuarios para que puedan registrarse.
Actor:	Administrador, entrenador

Tabla 5 Caso de uso app web 2 Generar links de registro



Nombre Caso de Uso:	Editar datos de usuarios
Identificador:	CS-03
Descripción:	El sistema debe permitir la gestión de los datos de los usuarios registrados.
Meta:	Permite que los administradores editen los datos como roles a los usuarios que ya existen.
Actor:	Administrador

Tabla 6 Caso de uso app web 3 Editar datos de usuario

Nombre Caso de Uso:	Consultar datos de perfil
Identificador:	CS-04
Descripción:	El sistema debe permitir la consulta de datos del perfil del usuario.
Meta:	Permite que los usuarios consulten los datos que ingresaron en su perfil en cualquier momento.
Actor:	Entrenador, esgrimista

Tabla 7 Caso de uso app web 4 Consultar datos de perfil

Nombre Caso de Uso:	Gestionar datos de perfil
Identificador:	CS-05
Descripción:	El sistema debe permitir el ingreso de datos de perfil tanto a usuarios nuevos como usuarios que deseen actualizar sus datos.
Meta:	Los usuarios podrán modificar sus datos o eliminarlos mientras no sean obligatorios.
Actor:	Entrenador, esgrimista

Tabla 8 Caso de uso app web 5 Gestionar datos de perfil

Nombre Caso de Uso:	Crear grupo de entrenamiento
Identificador:	CS-06
Descripción:	El sistema debe permitir al usuario crear un grupo de entrenamiento.
Meta:	Permite que los entrenadores creen un grupo de entrenamiento por arma.
Actor:	Entrenador

Tabla 9 Caso de uso app web 6 Crear grupo de entrenamiento



Nombre Caso de Uso:	Gestionar miembros de grupos
Identificador:	CS-07
Descripción:	El sistema debe permitir al usuario asignar miembros a cada grupo, y también eliminarlos.
Meta:	Permite que los entrenadores inserten usuarios esgrimistas a los grupos creados.
Actor:	Entrenador

Tabla 10 Caso de uso app web 7 Gestionar miembros de grupos

Nombre Caso de Uso:	Gestionar ciclos y planes de entrenamiento
Identificador:	CS-08
Descripción:	El sistema debe permitir la creación de ciclos de entrenamiento junto su planificación de ejercicios por ciclo.
Meta:	Permite que los entrenadores creen ciclos de entrenamiento en los cuales planifiquen entrenamientos cada día.
Actor:	Entrenador

Tabla 11 Caso de uso app web 8 Gestionar ciclos y planes de entrenamiento

Nombre Caso de Uso:	Generar objetivos
Identificador:	CS-09
Descripción:	El sistema debe permitir la generación de objetivos por ciclos.
Meta:	El entrenador podrá crear un objetivo para el ciclo de entrenamiento.
Actor:	Entrenador

Tabla 12 Caso de uso app web 9 Generar objetivos

Nombre Caso de Uso:	Visualizar entrenamientos de esgrimista
Identificador:	CS-10
Descripción:	El sistema debe permitir al usuario visualizar los entrenamientos individuales de cada esgrimista.
Meta:	Permite que los usuarios entrenadores puedan ver como los esgrimistas van progresando en sus ejercicios individuales.
Actor:	Entrenador



Tabla 13 Caso de uso app web 10 Visualizar entrenamientos de esgrimista

Nombre Caso de Uso:	Generar feedback
Identificador:	CS-11
Descripción:	El sistema debe permitir al usuario generar feedback a los entrenamientos que haya realizado el esgrimista y a los ciclos que hayan culminado
Meta:	Los entrenadores podrán retroalimentar a los esgrimistas en sus entrenamientos según su progreso.
Actor:	Entrenador

Tabla 14 Caso de uso app web 11 Generar feedback

Nombre Caso de Uso:	Visualizar planificación de entrenamiento
Identificador:	CS-12
Descripción:	El sistema debe permitir la visualización de la planificación de entrenamientos por ciclos al usuario.
Meta:	El esgrimista podrá ver la planificación de entrenamiento que se le ha asignado para el ciclo.
Actor:	Esgrimista

Tabla 15 Caso de uso app web 12 Visualizar planificación de entrenamiento

Nombre Caso de Uso:	Visualizar dashboard de entrenamientos
Identificador:	CS-13
Descripción:	El sistema debe permitir al usuario poder visualizar como ha progresado con los entrenamientos.
Meta:	El esgrimista podrá ver cómo ha progresado a lo largo de su planificación de ejercicios de una manera visual y gráfica.
Actor:	Esgrimista

Tabla 16 Caso de uso app web 13 Visualizar dashboard de entrenamientos

Nombre Caso de Uso:	Entrenar con módulo IA
Identificador:	CS-14
Descripción:	El sistema debe permitir al usuario acceder al módulo de IA a través de la aplicación web con una cámara.



Meta:	El esgrimista podrá realizar entrenamientos individuales a solas sin la supervisión de un entrenador para progresar en sus entrenamientos.
Actor:	Esgrimista

Tabla 17 Caso de uso app web 14 Entrenar con módulo IA

Nombre Caso de Uso:	Visualizar feedback y objetivos
Identificador:	CS-15
Descripción:	El sistema debe permitir al usuario visualizar todos los objetivos y el feedback recibido de los entrenamientos y ciclos.
Meta:	El esgrimista podrá recibir feedback manteniendo la interacción con los entrenadores para seguir mejorando con sus entrenamientos y conocer los objetivos del ciclo de entrenamiento.
Actor:	Esgrimista

Tabla 18 Caso de uso app web 15 Visualizar feedback y objetivos

Nombre Caso de Uso:	Registrar nuevo alumno
Identificador:	CS-16
Descripción:	El sistema debe permitir a los administradores y entrenadores registrar un nuevo alumno al sistema.
Meta:	Registrar un alumno como esgrimista para que pueda ingresar sus datos e ingresar al sistema mediante el login.
Actor:	Administradores y entrenadores

Tabla 19 Caso de uso app web 16 Registrar nuevo alumno

Nombre Caso de Uso:	Administrar entrenadores
Identificador:	CS-17
Descripción:	El sistema debe permitir a los administradores poder añadir y desactivar cuentas de entrenadores.
Meta:	Registrar nuevos entrenadores de la academia para que puedan ingresar sus datos, y en caso de requerirlo, desactivar sus cuentas en caso de dejar la academia.
Actor:	Administradores

Tabla 20 Caso de uso app web 17 Administrar entrenadores

Nombre Caso de Uso:	Cambiar incremento de puntos entre auto/manual
----------------------------	---



Identificador:	CM-1
Descripción:	La máquina de puntos debe permitir al usuario cambiar los puntos de manera automática con los toques o manualmente con un control remoto.
Meta:	Cambiar la forma en la que la máquina aumenta los puntos de los esgrimistas a manual o automática según las reglas del combate.
Actor:	Entrenadores

Tabla 21 Caso de uso máquina 1 Cambiar puntos entre automáticos y manual

Nombre Caso de Uso:	Sincronizar nuevo control
Identificador:	CM-2
Descripción:	La máquina de puntos debe ser capaz de sincronizar un nuevo control remoto.
Meta:	Sincronizar un nuevo control para que a la hora de encender la máquina esta pueda ser controlada a control remoto.
Actor:	Entrenadores

Tabla 22 Caso de uso máquina 2 Sincronizar nuevo control remoto

Nombre Caso de Uso:	Intercambiar puntaje
Identificador:	CM-3
Descripción:	La máquina de puntos debe ser capaz de intercambiar los puntajes entre sí.
Meta:	Cambiar los puntajes que se encuentran en lado izquierdo al lado derecho y viceversa.
Actor:	Entrenadores

Tabla 23 Caso de uso máquina 3 Intercambiar puntaje

Nombre Caso de Uso:	Reestablecer tarjetas de penalización
Identificador:	CM-4
Descripción:	La máquina debe remover las tarjetas amarillas y rojas que están activas.
Meta:	Remover las penalizaciones activas sin modificar los demás datos de la máquina



Actor:	Entrenadores
---------------	--------------

Tabla 24 Caso de uso máquina 4 Reestablecer tarjetas de penalización

Nombre Caso de Uso:	Asignar prioridad auto/manual
Identificador:	CM-5
Descripción:	La máquina de puntos debe ser capaz de asignar cuál de los dos lados del combate tiene prioridad.
Meta:	Asignar la prioridad de manera automática y aleatorio según las reglas del combate o manualmente con un control remoto.
Actor:	Entrenadores

Tabla 25 Caso de uso máquina 5 Asignar prioridad automática o manualmente

Nombre Caso de Uso:	Gestionar tarjetas amarillas/rojas
Identificador:	CM-6
Descripción:	La máquina de puntos debe gestionar las tarjetas amarillas y rojas del combate.
Meta:	Gestionar las tarjetas de cada jugador en el combate para llevar el conteo de penalizaciones en cada lado.
Actor:	Entrenadores

Tabla 26 Caso de uso máquina 6 Gestionar tarjetas amarillas y rojas

Nombre Caso de Uso:	Aumentar periodo
Identificador:	CM-7
Descripción:	La máquina de puntos debe ser capaz de cambiar el periodo en el que se encuentra el combate.
Meta:	Aumentar el periodo para determinar en qué parte del combate se encuentran actualmente los esgrimistas.
Actor:	Entrenadores

Tabla 27 Caso de uso máquina 7 Aumentar periodo

Nombre Caso de Uso:	Gestionar puntos
Identificador:	CM-8
Descripción:	La máquina de puntos debe ser capaz de editar los puntos de cada lado con un control remoto.



Meta:	Gestionar los puntos para que el árbitro pueda subir o bajar los puntos manualmente en caso de que las reglas del combate lo requieran.
Actor:	Entrenadores

Tabla 28 Caso de uso máquina 8 Gestionar puntos

Nombre Caso de Uso:	Pausar/continuar
Identificador:	CM-9
Descripción:	La máquina de puntos debe ser capaz de pausar el temporizador del combate y también proseguir con el tiempo.
Meta:	Pausar el tiempo en caso del que el árbitro lo requiera o algún esgrimista haya realizado un punto, y seguidamente poder continuar con el temporizador.
Actor:	Entrenadores

Tabla 29 Caso de uso máquina 9 Pausar o continuar temporizador

Nombre Caso de Uso:	Editar temporizador
Identificador:	CM-10
Descripción:	La máquina de puntos debe ser capaz de cambiar los valores de minutos, segundos, decimas y centésimas de segundo en el temporizador.
Meta:	Editar el temporizador a un tiempo determinado en caso de que un combate lo requiera.
Actor:	Entrenadores

Tabla 30 Caso de uso máquina 10 Editar temporizador

Nombre Caso de Uso:	Reestablecer
Identificador:	CM-11
Descripción:	La máquina de puntos debe ser capaz de reestablecer los puntos, tarjetas, periodo, prioridad y tiempo a su estado por defecto.
Meta:	Reestablecer todos los datos que tenga la máquina actualmente para preparar un siguiente combate.
Actor:	Entrenadores

Tabla 31 Caso de uso máquina 11 Reestablecer



7.1.3. Diseño y arquitectura

A. Aplicación web

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. 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. A continuación, se detalla cómo la solución está ideada en términos de diseño y arquitectura.

Siguiendo las descripciones de los requerimientos recolectados con la Academia, se logró determinar claramente que la aplicación debe contar con múltiples interfaces para mostrar e interactuar con distintos tipos de información de forma dinámica. Por esto, se optó por usar el patrón de diseño “Single Page Application” (SPA). Este patrón es ampliamente usado en el campo del desarrollo web y se ha vuelto popular últimamente en especial por la interacción rápida y fluida con el usuario y su relativa facilidad de implementación en múltiples plataformas debido a que el back-end y el front-end se encuentran desacoplados.

En una aplicación web tradicional, cada vez que la aplicación realiza una petición al servidor, el servidor responde con un nuevo documento HTML. Esto desencadena una actualización de página en el navegador. En contraste, en una aplicación SPA, después de que se carga la primera página, toda la interacción con el servidor ocurre a través de llamadas asíncronas (AJAX).

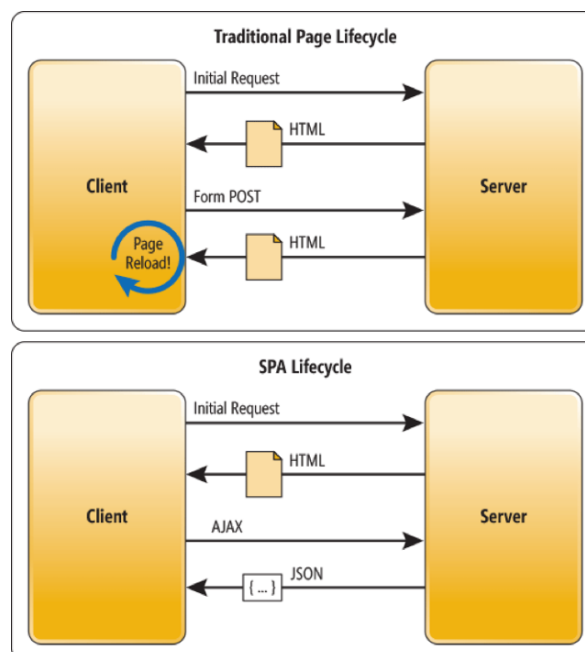


Diagrama 11 Ciclo de vida de una aplicación SPA (Microsoft, 2015)



Este tipo de aplicaciones brinda múltiples ventajas frente a una aplicación tradicional. En primer lugar, dado a que no existe la necesidad de entregar un documento HTML en cada petición los tiempos de respuesta mejoran considerablemente. En segundo lugar, debido a que existe una clara división de responsabilidades, el proceso de desarrollo y el mantenimiento de la aplicación son más eficientes ya que se pueden realizar cambios por separado. Adicionalmente, dado a que la lógica del servidor se encuentra desacoplada y funciona como un servicio web independiente, brinda la posibilidad de adaptar la aplicación a múltiples plataformas.

A un mayor nivel de detalle, en el back-end de la aplicación se optó por implementar una arquitectura por capas que contempla una capa de modelo que es la encargada de realizar operaciones con la base de datos. También cuenta con una capa de controladores los cuales son encargados de consumir la capa del modelo y gestionar las respuestas del servidor. El back-end expone un REST API, el cual es consumido por la aplicación del cliente y entrega la información procesada en formato JSON (JavaScript Object Notation).

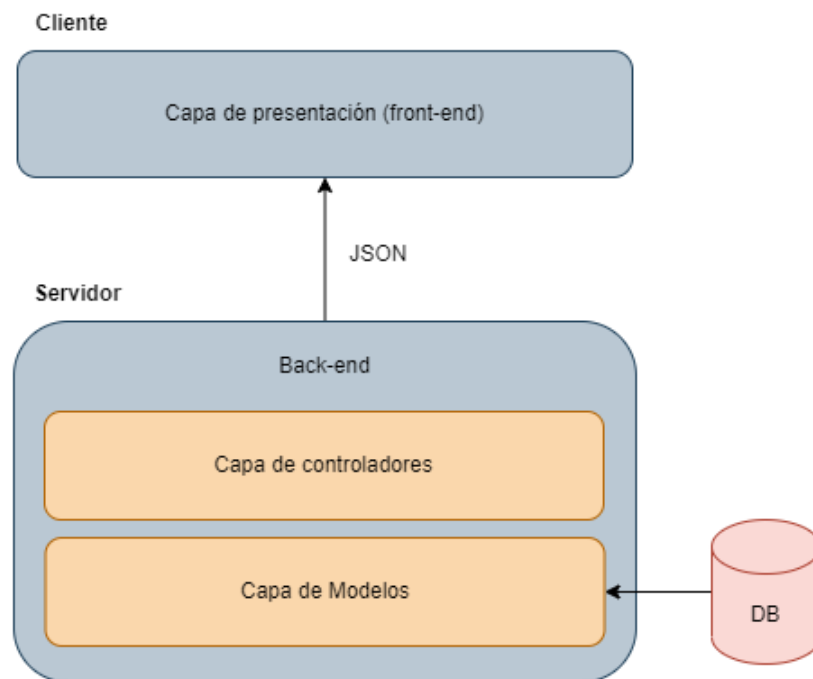


Diagrama 12 Arquitectura por capas de la aplicación web

Es necesario almacenar los datos generados por entrenadores y esgrimistas al usar la aplicación web. Dado que todos los datos que se necesitan manejar en la aplicación siguen una estructura predefinida y guardan relación entre sí (por ejemplo, entre el esgrimista al que se le asigna un plan o el entrenador que genera retroalimentaciones) se opta por utilizar una base de datos relacional. Tras el debido análisis de las entidades que representan los datos y sus relaciones, así como correcciones y nuevos



atributos a considerar, se representa el siguiente diagrama entidad-relación en una base de datos SQL.

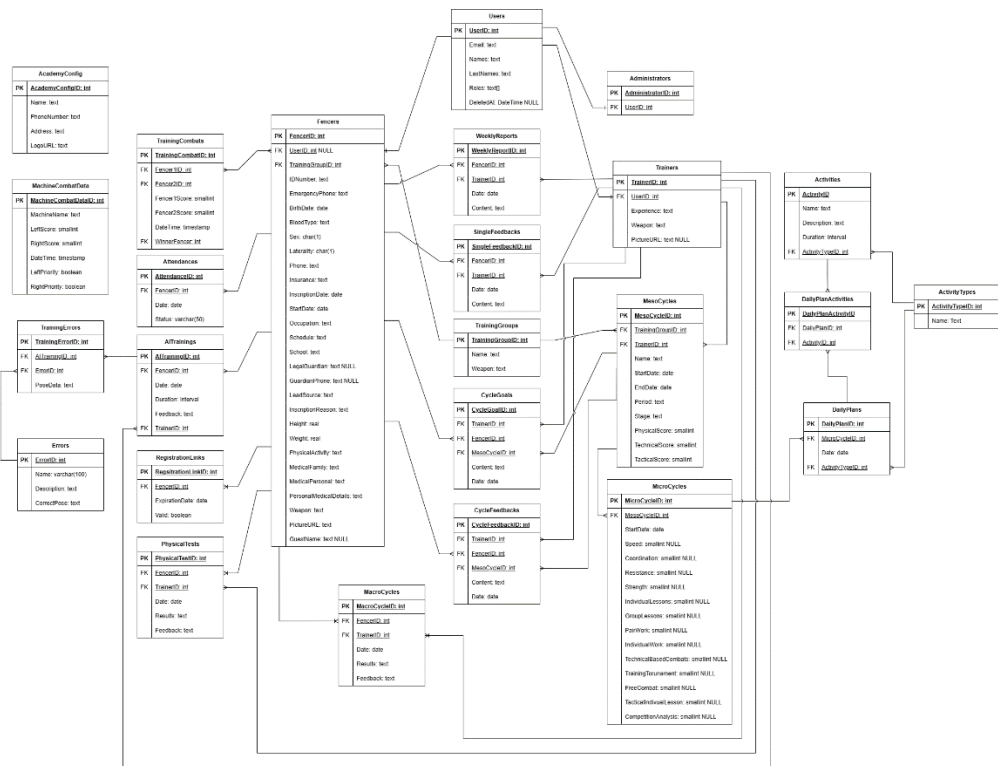


Diagrama 13 Diagrama entidad-relación del proyecto

B. Inteligencia artificial para detección de movimientos correctos

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 backend. Para integrar inteligencia artificial en esta aplicación web, será necesario utilizar o desarrollar una plataforma adecuada, que pueda ejecutarse en el entorno de web y tener funcionalidades de inteligencia artificial.

La captura de las poses en sí (es decir, la posición del esgrimista) se realizará en el frontend, 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 backend pueda analizar las mismas y determinar si el movimiento es correcto o se está potencialmente cometiendo un error. La detección



de poses correrá por un tiempo de pocos segundos para realizar el movimiento, todas las poses detectadas se enviarán al backend para análisis.

Una vez obtenidas las capturas de poses del frontend, entraría en funcionamiento la inteligencia artificial de análisis del backend. Este modelo debe ser construido y entrenado, pues no existe uno al que se pueda simplemente aplicar transfer learning debido a la especificidad de este caso. Al recibir los datos de las poses realizadas en un movimiento, esta inteligencia artificial determinará si el movimiento realizado coincide con los movimientos correctos aprendidos, y responderá al frontend indicando si es el caso o no. En caso de darse un error, esta ocurrencia se almacenará en la base de datos para que se la pueda visualizar al finalizar el entrenamiento.

El backend, tras completar su análisis, retornará una respuesta distinta al frontend dependiendo de si se encontró o no un error. En caso de que no se haya encontrado un error, el frontend leerá esto en la respuesta y le indicará al esgrimista que puede continuar realizando el movimiento para realizar nuevos análisis. Si un error fue detectado, la sesión de entrenamiento se pausa y se muestra al esgrimista el movimiento que realizó y cómo se debería ver el movimiento correcto. Adicionalmente, se pueden revisar los resultados de entrenamientos anterior con los datos almacenados.

Esencialmente se seguirá el flujo descrito a continuación y representado en el diagrama:

1. Captura de poses con inteligencia artificial pre entrenada usando la cámara del dispositivo. El proceso se lleva a cabo en el frontend de la aplicación.
2. Las poses capturadas (que comprenden el movimiento realizado, como cuadros de un video) se envían al backend para análisis.
3. La inteligencia artificial del backend (implementada específicamente para la solución) analiza las poses para determinar si el movimiento es correcto.
4. El backend retorna el resultado del análisis al frontend; el resultado indica si se detectó un error o no, en caso afirmativo, se retornan datos del movimiento realizado y el movimiento correcto se debería realizar.
5. Si el backend detectó un error, este se muestra al usuario con el movimiento que se debería realizar, caso contrario se permite continuar entrenando.

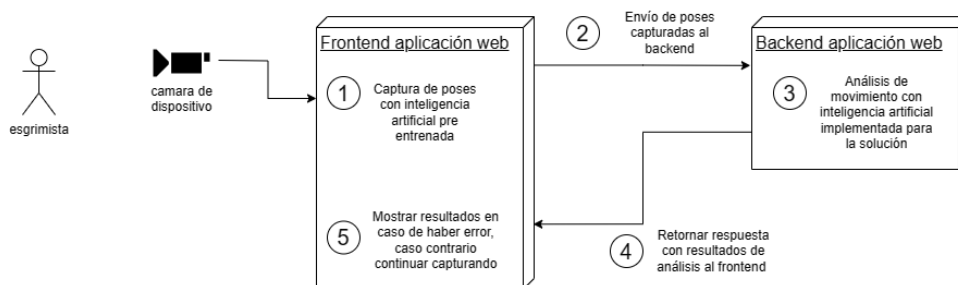


Diagrama 14 Flujo del funcionamiento de la inteligencia artificial para detección de posibles errores



C. Máquina de puntaje y temporizador

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, 2022b). Demás reglamentos de la FIE se deben considerar como funcionalidades básicas de la máquina. Los requerimientos de la Academia se han basado en tener las funcionalidades básicas con las que cuentan las máquinas oficiales, mismas que se han detallado anteriormente.

La máquina debe contar con displays para visualización, alertas sonoras, y un control remoto para permitir mejor interacción del usuario, mismos que se pueden controlar con microcontroladores. Los datos que se podrán visualizar en la máquina son el tiempo restante en el combate, puntajes, tarjetas de penalización, y prioridad (un indicador para cuando existen empates). Estos mismos podrán ser editados con ayuda del control remoto, de modo que el árbitro pueda utilizar la máquina desde su posición habitual a un costado de la pista pudiendo visualizar las acciones de los esgrimistas. Para aprovechar las funcionalidades de repetidor que tienen las máquinas existentes, se pueden leer los pulsos eléctricos de 12 voltios de esta funcionalidad para que la máquina de puntajes nueva pueda detectar cuando hay un toque y pausar el temporizador o subir puntajes según sea necesario.

La manera en la que la máquina se usaría durante un combate se puede visualizar gráficamente en el siguiente diagrama. Los esgrimistas, en sus posiciones predeterminadas en la pista, están conectados a la máquina de esgrima existente; junto a esta, se puede colocar la máquina de puntaje y temporizador de la solución, que el árbitro controla desde su posición con un control remoto.

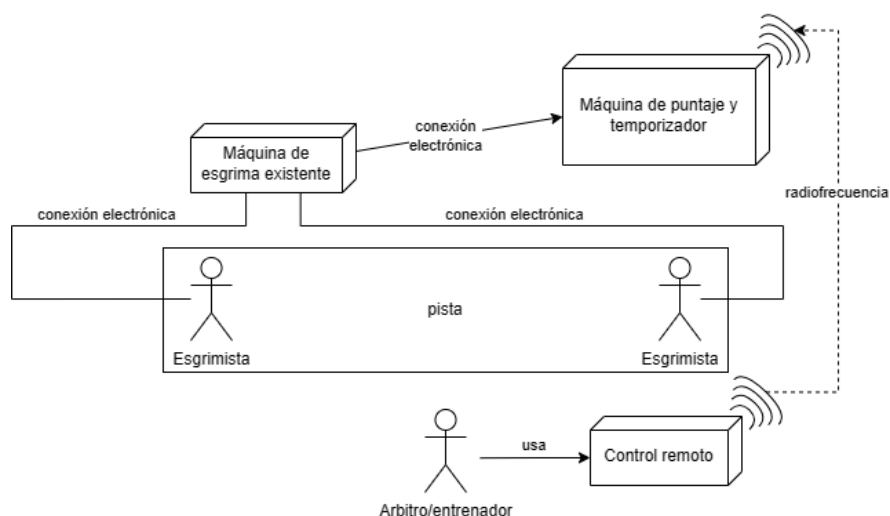


Diagrama 15 Representación de la máquina de puntaje siendo usada en un combate



D. Vista general de la solución

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 en la nube. 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, y se comunican entre sí por radio frecuencia. Los siguientes diagramas pueden asistir a entender la manera en que la solución está construida.

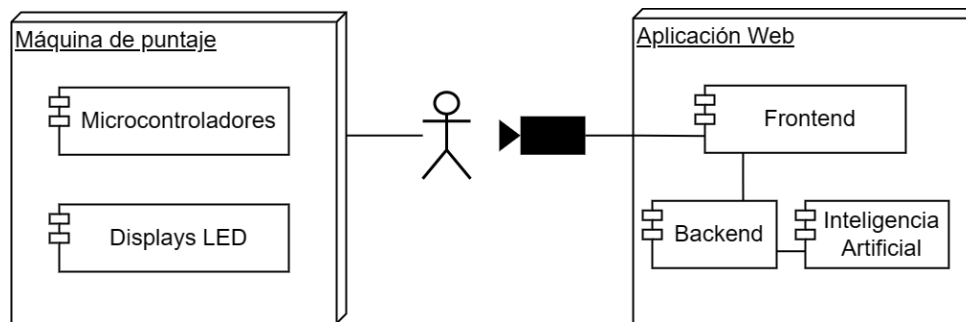


Diagrama 16 Representación simplificada de la solución

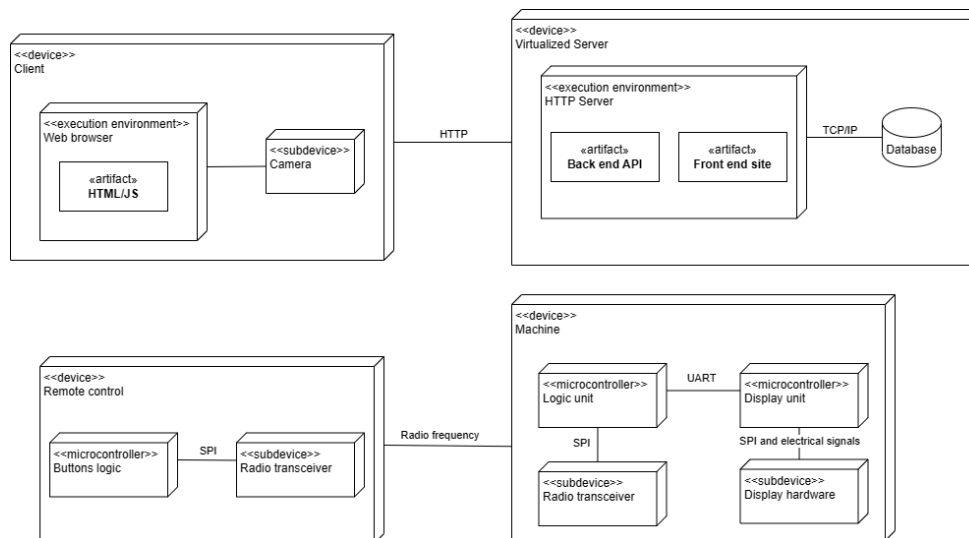


Diagrama 17 Diagrama de despliegue de la solución

Para representar la solución a un nivel más detallado, se puede considerar el siguiente diagrama de componentes. El mismo contiene la estructura general del código de la aplicación web (backend y frontend) y del software que se ejecuta en la máquina de puntaje; así como comunicaciones entre componentes. Con este diagrama se tiene una vista más técnica de la solución, haciendo ya mención a librerías y componentes de código específicos que permiten lograr las funcionalidades, diseño, y arquitectura planteadas.

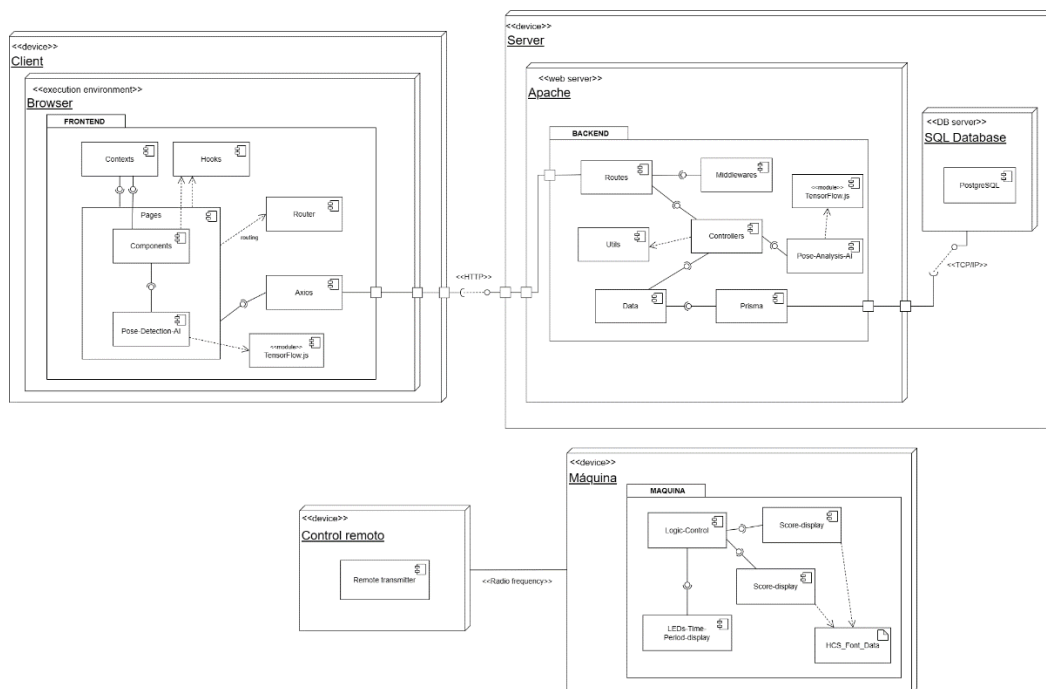


Diagrama 18 Diagrama de componentes del proyecto

7.1.4. 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. Para ello la recopilación de requisitos, además del inicio del proyecto, se han recogido periódicamente con cada avance del desarrollo. Esto implica reuniones periódicas con los stakeholders, en las cuales se recopilan sus necesidades para la solución, se validan las representaciones de sus necesidades en el diseño del sistema y máquina, y se realizan correcciones según sea necesario.

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. Los sprints que se planificaron para el desarrollo, así como los productos resultantes de cada uno e historias de usuario relacionadas, se listan continuación:

- Sprint 1: 2023-02-15 a 2023-03-15
 - Productos resultantes:
 - Determinar componentes y precios para la máquina



- Determinar que errores detectar con la IA
- Preparar el ambiente de trabajo del frontend y backend
- Crear diagrama ER de la base de datos
- Historias de usuario:
 - Como entrenador de la academia deseo tener una máquina de puntaje que sea capaz de mostrar los datos del combate actual para no llevar la cuenta mentalmente de los puntos del combate.h
- Sprint 2: 2023-03-16 a 2023-04-16
 - Productos resultantes:
 - Diseño del control remoto
 - Prototipo del display de la máquina de puntos
 - Wireframes de planes de entrenamiento, entrenadores y creación de usuarios
 - Crear la base de datos
 - Historias de usuario:
 - Como entrenador o administrador de la academia deseo poder registrar un nuevo usuario a la academia para que el mismo pueda ingresar con sus credenciales al sistema de la academia.
 - Como entrenador de la academia deseo poder generar ciclos y planes de entrenamiento para determinar que actividades deben hacer los esgrimistas en cada día.
 - Como entrenador de la academia deseo visualizar los entrenamientos individuales de los esgrimistas para poder ver cómo están progresando con sus actividades.
 - Como administrador de la academia deseo poder gestionar entrenadores para poder modificar sus datos y desactivar su acceso al sistema en caso de dejar la academia.
- Sprint 3: 2023-04-17 a 2023-05-01
 - Productos resultantes:
 - Demo del display de la máquina de puntos
 - Código del control y set up
 - Prototipo de detección de poses con TensorFlow
 - Escoger modelo de TF.js
 - Wireframes de esgrimistas
 - Login y registro en frontend
 - CRUD y endpoints en backend
 - Historias de usuario:
 - Como entrenador y esgrimista de la academia debo poder ver los datos de mi perfil para consultar los datos con los que me haya registrado.



- Como usuario de la plataforma quiero ser capaz de ingresar al sistema con una cuenta personal para poder almacenar mis datos y acceder a mis funcionalidades dentro del sistema.
- Como entrenador y esgrimista de la academia deseo poder cambiar mis datos de perfil, médicos y personales en cualquier momento en caso de que mis datos oficiales cambien.
- Como administrador de la academia deseo tener la posibilidad de editar los datos de los usuarios en caso de que ellos olviden algún dato necesario para ingresar al sistema.
- Como entrenador y esgrimista de la academia debo poder ver los datos de mi perfil para consultar los datos con los que me haya registrado.
- Como entrenador y esgrimista de la academia deseo poder cambiar mis datos de perfil, médicos y personales en cualquier momento en caso de que mis datos oficiales cambien.
- Sprint 4: 2023-05-02 a 2023-05-15
 - Productos resultantes:
 - Probar distancia del control remoto
 - Sketch y esquemas de conexión de la máquina y el control
 - Display del puntaje
 - Implementar red neuronal en backend para IA
 - Diagrama de componentes y despliegue del sistema
 - Wireframes de IA
 - Pipelines de CD para la aplicación web
 - Historias de usuario:
 - Como esgrimista de la academia deseo realizar mis entrenamientos con una inteligencia artificial que me asista en mis movimientos sin la necesidad de tener un entrenador observándome.
 - Como entrenador deseo poder cambiar la forma de incrementar los puntos de la máquina de forma automática mediante toques o de forma manual mediante un control remoto para mantener el conteo de puntos durante un combate.
 - Como entrenador deseo poder ser capaz de editar los puntos de cada lado a través de un control remoto para incrementar o disminuir el puntaje de forma manual en caso de que las reglas del combate lo requieran.
- Sprint 5: 2023-05-15 a 2023-05-29
 - Productos resultantes:
 - Incrementar el rango del control remoto y su recepción
 - Lógica y botones de auto puntaje y sincronización de control remoto en máquina de puntaje
 - Logica de toques en máquina de puntaje



- Esquema de display de puntaje y tiempo
- Generación de JSON de poses con video
- Implementación de IA en backend
- Grupos, detalles de grupos, registro de esgrimista, generación de links de invitación y listado de esgrimistas en frontend
- Historias de usuario:
 - Como entrenador de la academia deseo gestionar los miembros de los grupos creados para poder ingresar o eliminar esgrimistas en caso de que cambien su arma, salgan de la academia o cambien de grupo.
 - Como entrenador deseo poder sincronizar un nuevo control remoto para que a la hora de encender la maquina esta pueda ser controlada de forma remota.
 - Como entrenador deseo poder ser capaz de pausar y reanudar el temporizador del combate a través de un control remoto para mantener un conteo de tiempo restante preciso durante un combate.
 - Como entrenador de la academia deseo crear grupos de entrenamiento para clasificar los grupos de esgrimistas en la academia por el arma que usen y la descripción de su grupo.
 - Como administrador y entrenador de la academia deseo crear links de invitación para enviar a nuevos esgrimistas y que puedan registrarse en la aplicación.
 - Como entrenador deseo poder sincronizar un nuevo control remoto para que a la hora de encender la maquina esta pueda ser controlada de forma remota.
 - Como entrenador o administrador de la academia deseo poder registrar un nuevo alumno a la academia para que el mismo pueda ingresar con sus credenciales al sistema de la academia.
 - Como entrenador deseo poder cambiar la forma de incrementar los puntos de la máquina de forma automática mediante toques o de forma manual mediante un control remoto para mantener el conteo de puntos durante un combate.
- Sprint 6: 2023-05-30 a 2023-06-12
 - Productos resultantes:
 - Implementación de comandos a control remoto.
 - Recablear control remoto a Arduino MEGA
 - Arreglo de bugs en control remoto
 - Endpoints de IA en backend
 - Feedback, mesociclo y planificación de entrenamientos en frontend



- Historias de usuario:
 - Como entrenador de la academia deseo poder generar ciclos y planes de entrenamiento para determinar que actividades deben hacer los esgrimistas en cada día.
 - Como esgrimista de la academia deseo visualizar la planificación de entrenamiento de los ciclos establecidos para conocer qué actividades debo realizar cada día.
 - Como entrenador deseo poder intercambiar los puntajes de lado izquierdo al lado derecho y viceversa para manejar cambios de lado en la pista
 - Como entrenador deseo poder ser capaz de reestablecer los valores de la maquina a su estado por defecto para prepararla para un siguiente combate.
 - Como entrenador deseo poder ser capaz de cambiar los valores de minutos, segundos, decimas y centésimas de segundo del temporizador para editar el tiempo restante de un combate en caso de que el árbitro lo considere necesario.
 - Como entrenador deseo poder ser capaz de cambiar el periodo en el que se encuentra el combate para mantener el conteo del periodo visible para los participantes del combate.
 - Como entrenador deseo poder gestionar tarjetas de amarillas y rojas para mantener el conteo de penalizaciones de los esgrimistas que participan en un combate.
 - Como entrenador deseo poder asignar cuál de los dos lados del combate tiene prioridad de forma automática o aleatoria para adaptarse a las reglas del combate definidas.
 - Como entrenador deseo poder reestablecer las tarjetas de penalización amarillas y rojas que se encuentren activas para manejar desacuerdos durante un combate.
 - Como entrenador de la academia deseo poder generar un feedback esporádico a los esgrimistas para enviarles retroalimentación de sus progresos en los entrenamientos.
 - Como esgrimista de la academia deseo poder visualizar los feedbacks y objetivos de un mesociclo específico asignado por mi entrenador para conocer el objetivo y la retroalimentación de mi desempeño de un ciclo de entrenamiento.
 - Como esgrimista de la academia deseo visualizar la planificación de entrenamiento de los ciclos establecidos para conocer qué actividades debo realizar cada día.
- Sprint 7: 2023-06-13 a 2023-06-26
 - Productos resultantes:
 - Esquema de unidad lógica de la máquina de puntaje
 - Pruebas y arreglo de bugs en máquina de puntaje



- Grabar videos de movimientos correctos para entrenamiento de IA
- Combates y vista de grupos para esgrimistas en frontend
- Arreglo de bugs en frontend
- Actualizar ER de la base de datos
- Historias de usuario:
 - Como entrenador deseo poder intercambiar los puntajes de lado izquierdo al lado derecho y viceversa para manejar cambios de lado en la pista.
 - Como entrenador deseo poder reestablecer las tarjetas de penalización amarillas y rojas que se encuentren activas para manejar desacuerdos durante un combate.
 - Como arbitro de combate deseo poder asignar cuál de los dos lados del combate tiene prioridad de forma automática o aleatoria para adaptarse a las reglas del combate definidas.
 - Como entrenador deseo poder gestionar tarjetas de amarillas y rojas para mantener el conteo de penalizaciones de los esgrimistas que participan en un combate.
 - Como entrenador deseo poder ser capaz de cambiar el periodo en el que se encuentra el combate para mantener el conteo del periodo visible para los participantes del combate.
 - Como entrenador deseo poder ser capaz de reestablecer los valores de la maquina a su estado por defecto para prepararla para un siguiente combate.
 - Como entrenador de la academia deseo gestionar los miembros de los grupos creados para poder ingresar o eliminar esgrimistas en caso de que cambien su arma, salgan de la academia o cambien de grupo.
- Sprint 8: 2023-06-27 a 2023-07-14
 - Productos resultantes:
 - Cubierta de la máquina de puntaje
 - Arreglo de bugs de la maquina
 - Ensamblar control remoto
 - Guardado de datos de IA en base de datos
 - Reentrenar modelo con data augmentation
 - Configuración de la academia, actividades, tipo de actividades, feedback y metas del ciclo, y, entrenamiento de IA en frontend, objetivos y feedbacks de mesociclo.
 - Añadir movimientos de IA en frontend
 - Arreglo de bugs frontend
 - Historias de usuario:
 - Como entrenador deseo poder ser capaz de editar los puntos de cada lado a través de un control remoto para incrementar o disminuir el puntaje de forma manual en caso de que las reglas del combate lo requieran.



- Como entrenador deseo poder ser capaz de pausar y reanudar el temporizador del combate a través de un control remoto para mantener un conteo de tiempo restante preciso durante un combate.
- Como entrenador deseo poder ser capaz de cambiar los valores de minutos, segundos, decimas y centésimas de segundo del temporizador para editar el tiempo restante de un combate en caso de que el árbitro lo considere necesario.
- Como esgrimista de la academia deseo poder visualizar los feedbacks y objetivos de un mesociclo específico asignado por mi entrenador para conocer el objetivo y la retroalimentación de mi desempeño de un ciclo de entrenamiento.
- Como entrenador de la academia deseo poder generar objetivos que lograr en los ciclos de entrenamiento para que los esgrimistas tengan conocimiento de las metas que deben alcanzar.
- Como esgrimista de la academia deseo visualizar un dashboard de mi progreso en los entrenamientos para tener una vista general de que tan bien me ha ido en las actividades del ciclo.

Para el desarrollo del proyecto, se requieren habilidades con inteligencia artificial, JavaScript, diseño web, bases de datos, Arduino (C++) y electrónica. Se ha determinado que, según las habilidades y aptitudes de cada integrante del proyecto, se pueden designar líderes para el desarrollo de distintas funcionalidades según las habilidades requeridas. Esto, sumado a que el proyecto posee tres partes principales que desarrollar, resulta en que se opte por separar el liderazgo que tiene cada integrante a cada 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. Para cumplir con el desarrollo del proyecto, en algún momento será necesario adquirir nuevas habilidades técnicas o conocimientos acerca de componentes y librerías. Esto significa que, conforme sea necesario, se usarán: tutorías con el docente guía,



consultas en línea o en bibliotecas, y consultas con profesores del campo adecuado o expertos.

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.

A nivel de 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.

Cuando sea necesario evaluar cambios, o tomar de decisiones de impacto para el proyecto, será necesario evaluarlas a nivel de equipo, y con el docente guía de ser necesario para contar con su visión de experto. Esencialmente, lo que se busca es que, de presentarse un punto de decisión importante para el proyecto, se siga priorizando la comunicación interna y el análisis detallado de los factores involucrados. Se seguirían los siguientes pasos:

- Identificar el problema a resolverse, los factores, causas y necesidad de resolución.
- Determinar alternativas de solución (opciones) de las cuales se debe seleccionar una solución definitiva.
- En conjunto con el equipo, identificar los puntos fuertes y débiles de las alternativas, de modo que se puedan evaluar para tomar una decisión. De ser necesario, se pueden considerar más alternativas que surjan del análisis. Si se considera necesario para tomar una decisión más informada, se debe informar al docente guía de las opciones en una de las reuniones periódicas con el mismo para obtener apoyo.
- A partir de análisis previo, definir la mejor solución o alternativa al problema que debe ser solucionado.
- Implementar la alternativa seleccionada, ya sea a nivel individual o de equipo.

7.2. Desarrollo de la solución

7.2.1. Ambiente de desarrollo

Para el desarrollo de la solución se requieren los siguientes elementos de infraestructura:

- Hardware:
 - Tres computadoras con 8GB de RAM mínimo
 - Conexión a internet



- Protoboards (para máquina de puntaje y control remoto)
- Suministros de poder de 5V y 12V DC (para máquina)
- Baterías (2 packs de 4xAA, para máquina y control remoto)
- Software:
 - IDEs (Visual Studio Code, Vim, Arduino IDE)
 - Controlador de versionamiento (git)
 - Base de datos PostgreSQL
 - Yarn y NodeJS
 - Navegador web (basado en Chromium)
 - Windows 10 o superior para computadoras de desarrollo
 - Ubuntu (Desktop o server) para servidor de pruebas
 - Servidor Web (Apache HTTP Server)

Como plan de mantenimiento de la infraestructura para el desarrollo, se establece que se debe mantener las herramientas usadas siempre actualizadas a la versión estable más reciente. En el caso de PostgreSQL en la versión 15, NodeJS en la versión 18 y los navegadores en su última versión disponible.

7.2.2. Implementación

Adicionalmente a las descripciones técnicas de implementación se pueden encontrar archivos de diseño, código fuente, y esquemáticas de conexión en el repositorio de GitHub del proyecto: [Diego-Hiriart/Capstone Hiriart-Corales-Samaniego: Diego Hiriart, Luis Corales and Christian Samaniego's capstone project por Software Engineering - UDLA \(github.com\)](https://github.com/Diego-Hiriart/Capstone-Hiriart-Corales-Samaniego).

A. Aplicación web

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. Se optó por utilizar Material UI para acelerar el tiempo de desarrollo, la cual es una librería de componentes y utilidades para React que facilitan la creación de interfaces de usuario con diseño responsive.

Para la creación inicial del frontend se utilizó Vite, la cual permite construir de forma optimizada la aplicación de React junto con sus dependencias y adicionalmente correr servidores de desarrollo. Para 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 el servidor.

La capa de presentación (frontend) 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. A continuación, se detalla la estructura de la capa de presentación (frontend):

- **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 desarrolló el backend para la obtención y registro de datos en la base de datos. Para ello se inicializo Prisma, una librería ORM (Object Relational Mapping) que adicionalmente fue utilizada para la creación de la base de datos en PostgreSQL 15.

Para hacer la conexión entre el backend y el frontend se desarrolló un REST API con el framework web Express. 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 frontend, la maquina y el módulo de inteligencia artificial que trabajan por separado, pero al final de cumplir sus funciones todas deben apuntar al backend 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.

Se desarrollaron diagramas de las funcionalidades más importantes del sistema, entre ellas se encuentran:

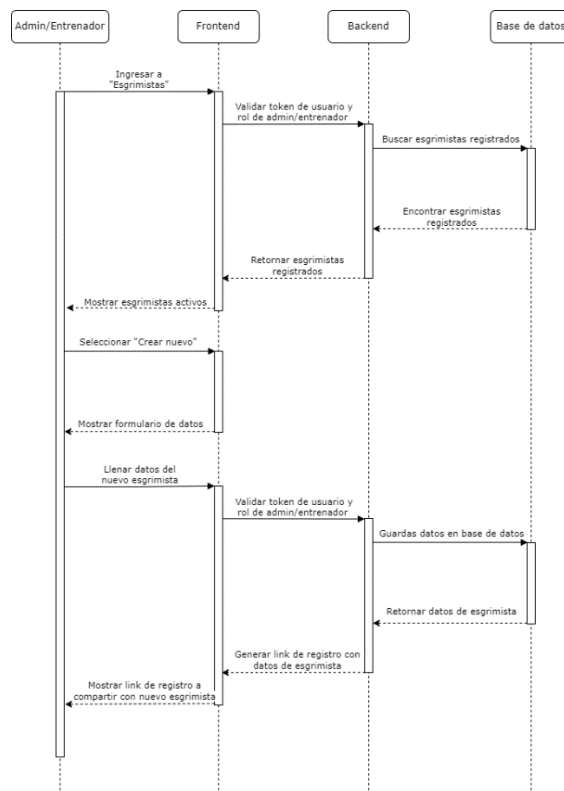


Diagrama 19 de secuencia de funcionalidad crear link de registro de nuevo esgrimista



Una de las funcionalidades principales del sistema requerido para la academia es el registro de nuevos esgrimistas. Para lograr esta funcionalidad se creó un proceso de invitación a nuevos usuarios. Esto consiste en que un usuario de rol administrador o entrenador ingrese a la respectiva página en la aplicación web para generar un link de invitación. Este link es único ya que en el URL generado se añade un token que valida el usuario a registrarse, guardando el email del nuevo esgrimista en la base de datos y cuando el nuevo usuario se vaya a registrar deba ingresar el mismo email. Esta funcionalidad se completa con el siguiente proceso:

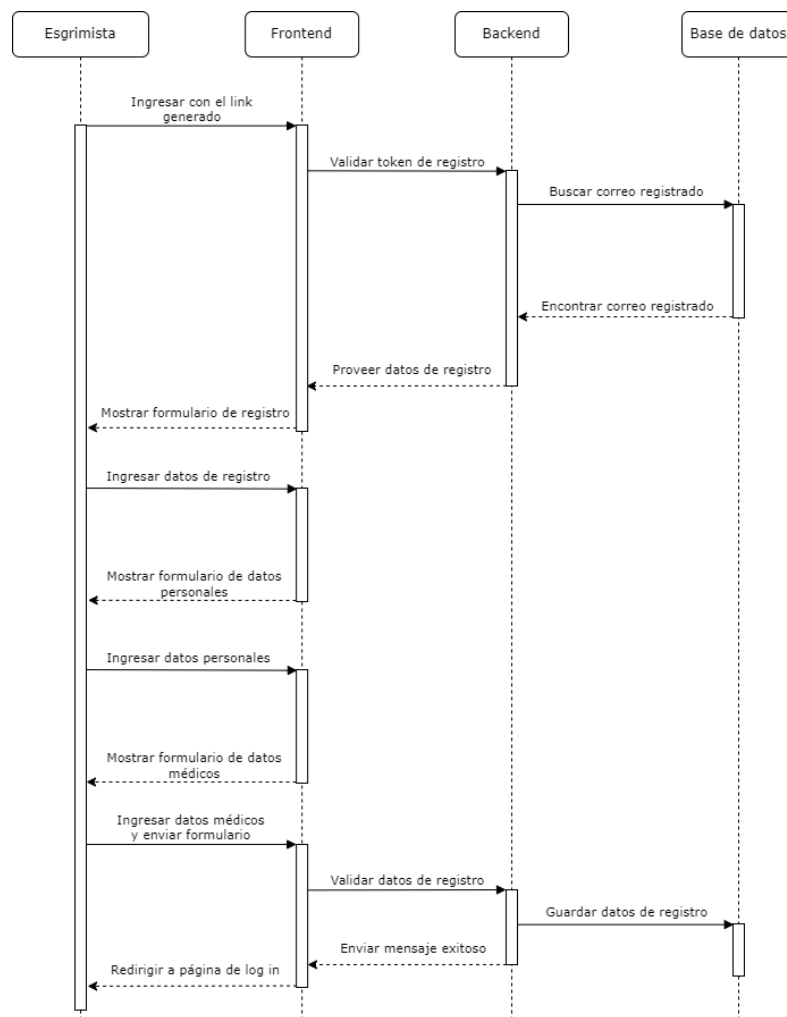


Diagrama 20 Diagrama de secuencia de funcionalidad registro de nuevo esgrimista

El nuevo esgrimista recibirá el link generado anteriormente para poder ingresar a la pantalla de registro, y allí tendrá que llenar un formulario con los mismos datos que se usaron para generar el link. Seguidamente, deberá ingresar sus datos personales y finalmente sus datos médicos (todos son necesarios por temas de seguridad en la academia). Al enviar el formulario, se registrará el usuario con los datos ingresados y será redirigido a la pantalla de log in.

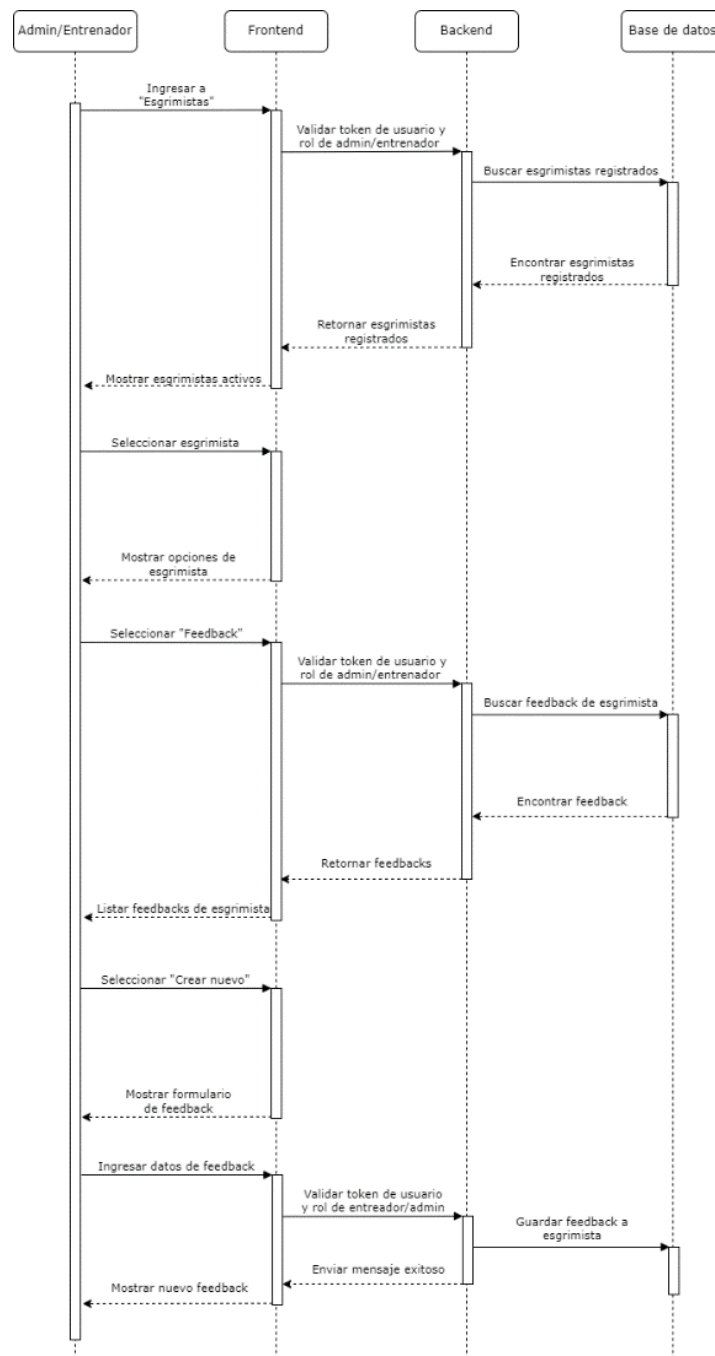


Diagrama 21 Diagrama de secuencia de funcionalidad feedback esporádico a esgrimista

Otras de las funcionalidades principales del sistema es la posibilidad de proveer feedback esporádico a todos los integrantes de la academia. Anteriormente, el feedback que se podía proveer a los esgrimistas solo quedaba en un encuentro dentro de la academia, pero gracias a la aplicación web ahora se podrá crear un registro tangible el cual el esgrimista podrá acceder en cualquier momento.

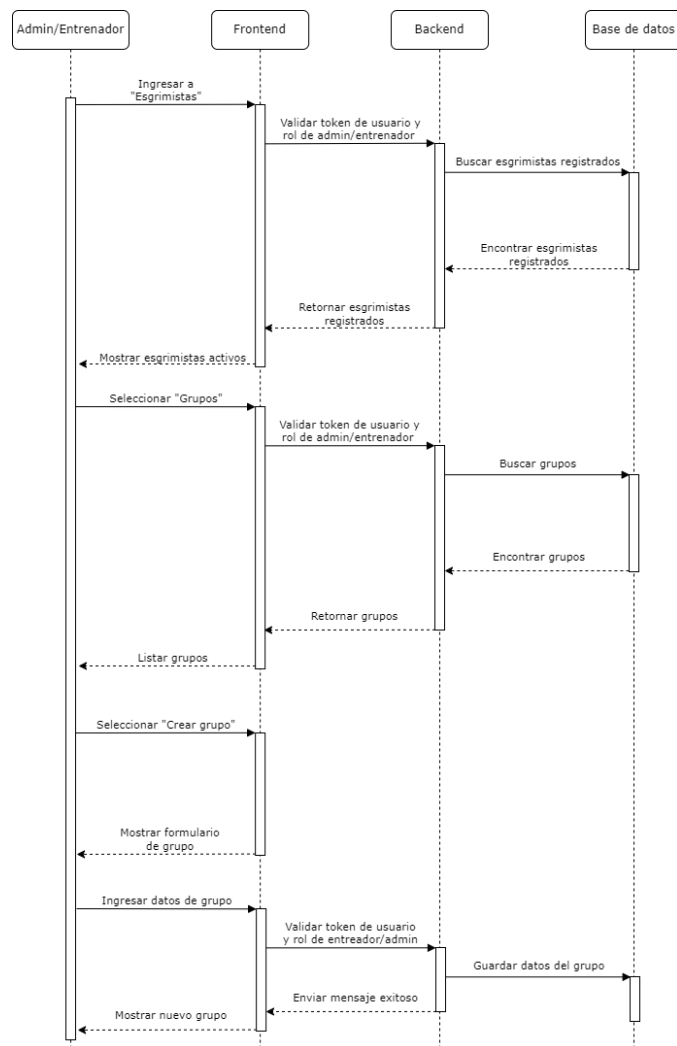


Diagrama 22 Diagrama de secuencia de funcionalidad crear un nuevo grupo

Los entrenadores podrán organizar todos los esgrimistas de la academia en grupos clasificados por nombres y el arma usada. Tanto los entrenadores, que pueden ver todos los grupos existentes y sus integrantes, como los esgrimistas que pueden ingresar a los grupos a los que pertenecen, podrán ver todas las opciones dentro del grupo. Visualizar el feedback y metas del grupo, los demás integrantes y los mesociclos junto con sus planificaciones de entrenamiento.

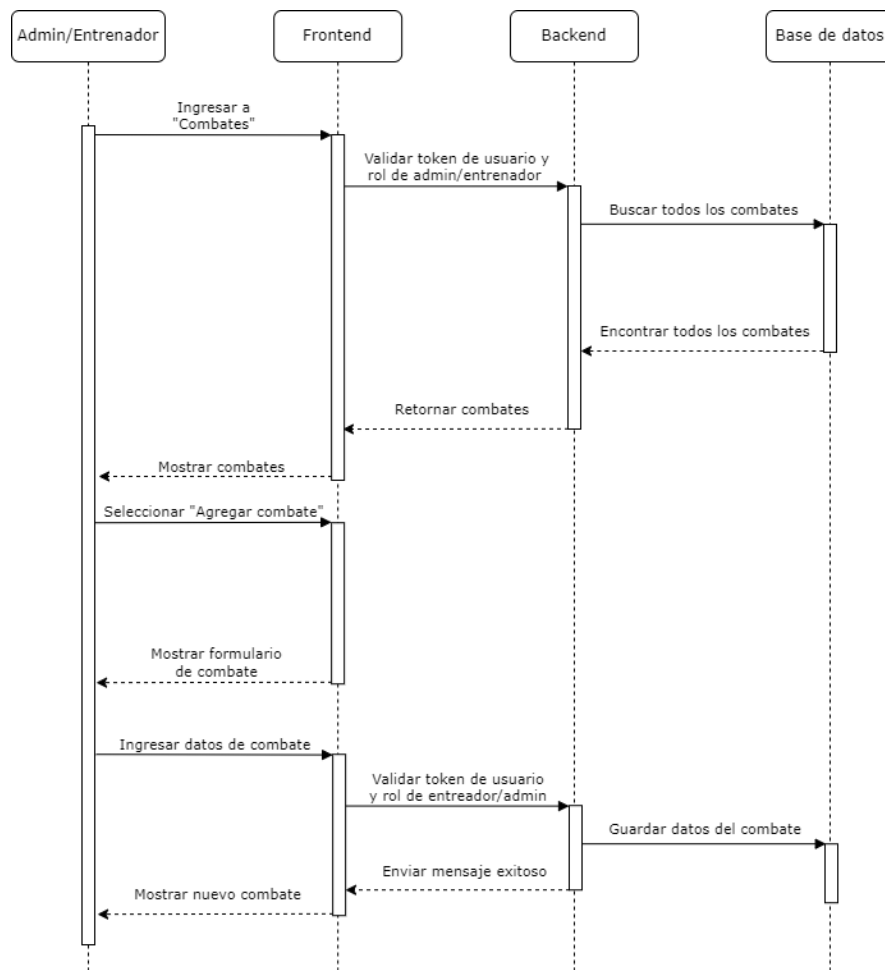


Diagrama 23 Diagrama de secuencia de secuencia de funcionalidad registrar combate

Es posible tener un registro de todos los combates, para ello, un usuario de rol administrador o entrenador podrá ingresar a la pestaña de combates y visualizar todos los combates registrados. También podrá añadir un nuevo combate ingresando datos como la fecha, los esgrimistas participantes y el puntaje de cada uno. Esto permite que los entrenadores puedan tener una visualización de los combates más recientes, y los esgrimistas poder ver los combates en los que hayan participado.

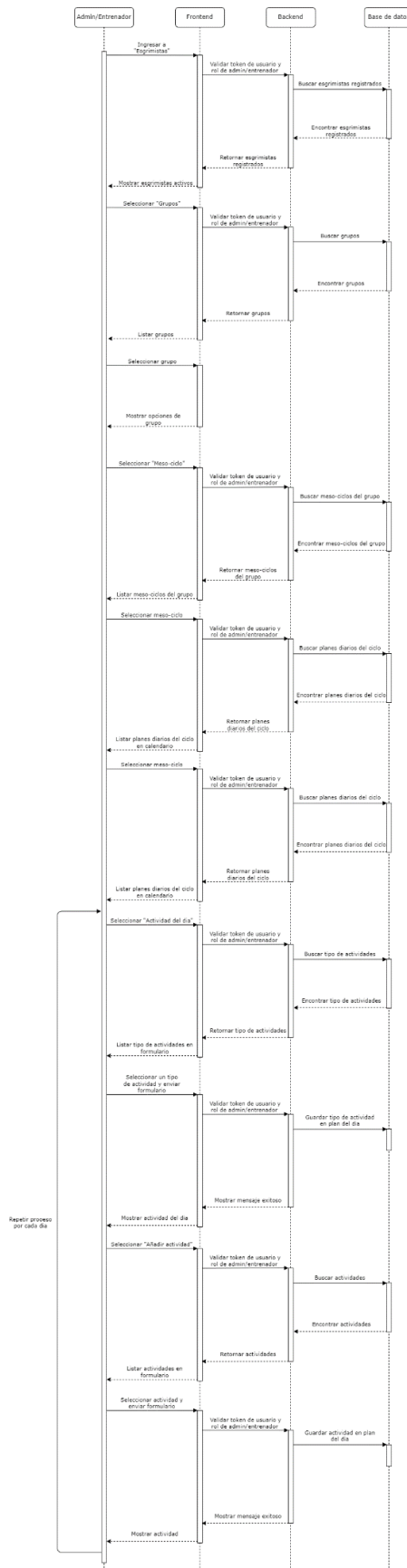


Diagrama 24 Diagrama de secuencia de funcionalidad planificación del día



En cada mesociclo se genera automáticamente una serie de microciclos que cada día perteneciente a ellos tendrá un plan de entrenamiento personalizable. El entrenador o administrador podrá ingresar a un mesociclo e ingresar la actividad del día y las actividades de ese día para que seguidamente los esgrimistas puedan ver que planificación de entrenamiento tienen.

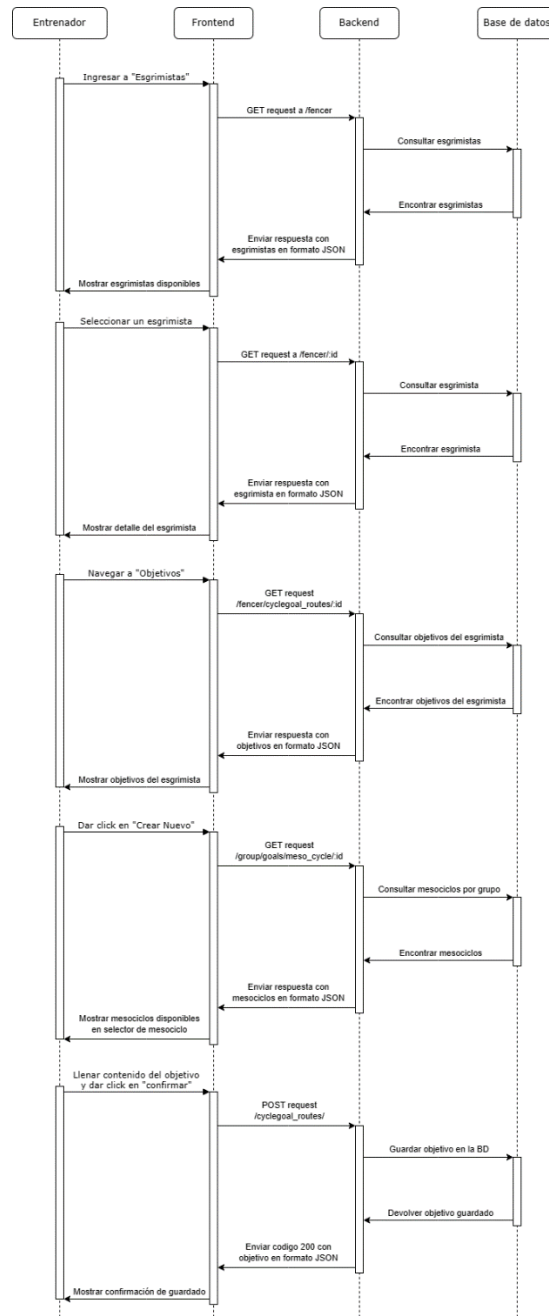


Diagrama 25 Diagrama de secuencia de funcionalidad de adición de objetivo de un mesociclo.

Un entrenador puede asignar un objetivo a cada esgrimista por mesociclo. De esta forma un entrenador puede definir el alcance personalizado de un mesociclo cuando



el mismo ha iniciado y este objetivo podrá ser consultado por el esgrimista en cualquier momento.

B. Inteligencia artificial para entrenamiento

La inteligencia artificial de detección de posibles errores, implementada con la librería TensorFlow.js, está integrada tanto en el frontend como backend de la aplicación web. La porción de este módulo que se ejecuta en el frontend (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. Estas poses se capturan durante 3 segundos, y son entonces se envían al backend para ser analizadas 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. Debido al consumo de recursos para ejecutar la detección de poses de forma continua, esta funcionalidad es mejor aprovechada en un computador o dispositivo móvil de alta gama.

El análisis de las poses es realizado por un modelo creado específicamente para este propósito, usando TensorFlow para generar una red neuronal tipo Multi Layer Perceptron (MLP). Este es uno de los tipos más básicos de redes neuronales artificiales, que consiste en múltiples capas de neuronas (filas o columnas, según el sentido de representación). Estas capas permiten el procesamiento de datos de entrada para obtener un resultado, que en este caso es la determinación de la clase o tipo de dato que se ingresó. Su uso para clasificación es óptimo debido a que permite separar puntos de datos de manera no lineal (Bento, 2021). Siendo un modelo de clasificación, cuando el frontend envía el conjunto de poses, la inteligencia artificial del backend determina si el movimiento realizado es correcto o se trata de un error. La inteligencia artificial da como salida un arreglo de valores que indica en porcentaje a que clase pertenece el movimiento analizado. Este arreglo contiene porcentajes de certeza para todas las clases, siendo el más alto el que el modelo considera que se ha detectado. Las clases que se pueden detectar son los movimientos correctos; si la certeza del modelo de que los datos analizados pertenezcan a la clase del movimiento correcto es muy baja (o considera que es otra clase) se notifica que se detectó un error. En el caso de detectarse un error, se guarda el mismo en la base y se lo retorna junto con el movimiento correcto. El proceso completo de detección y una representación de la topología del modelo se muestran a continuación.

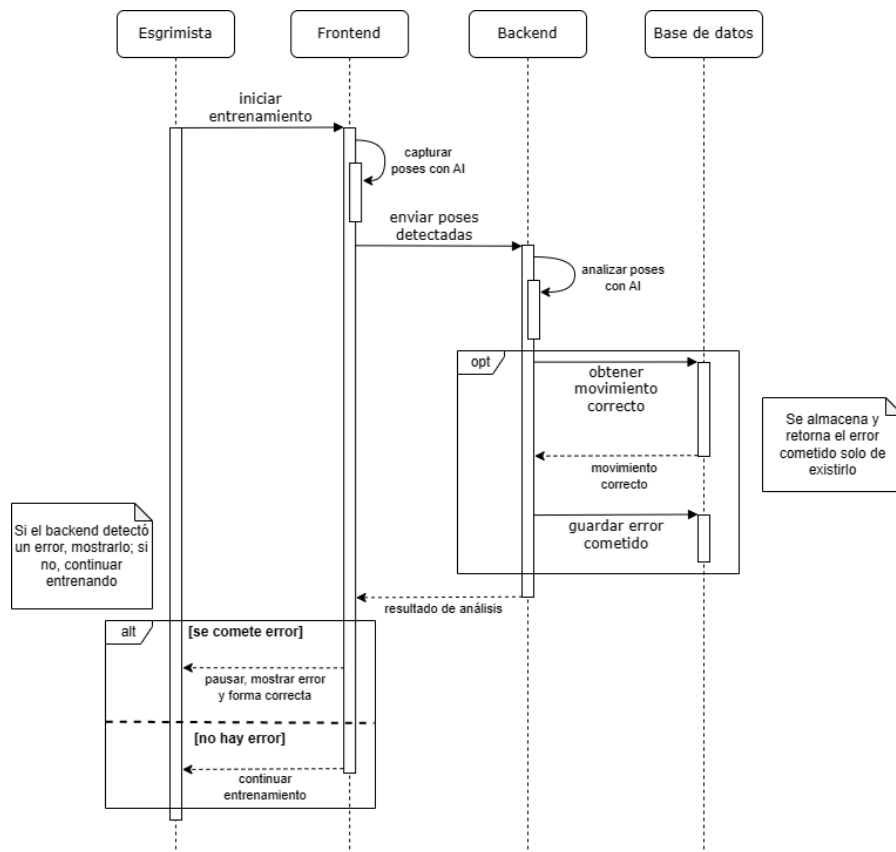


Diagrama 26 Diagrama de secuencia del uso de la funcionalidad de entrenamiento con AI

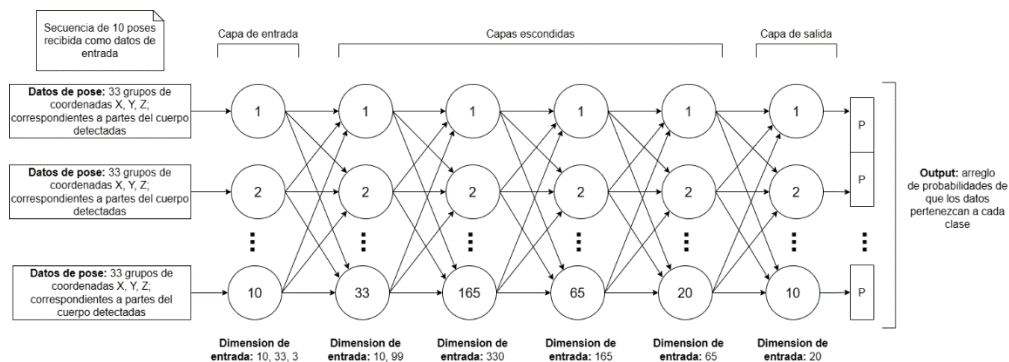


Diagrama 27 Representación de la topología del MLP usado para análisis

Para entrenar el modelo de análisis de movimientos, se generó un dataset de datos capturados con esgrimistas y con data augmentation. La captura de datos inició con grabaciones de los esgrimistas más experimentados de la academia realizando los movimientos que se quieren poder clasificar. Estos videos luego son procesados, para obtener clips cortos de cada movimiento, vigilando que sean datos representativos. Estos clips de video luego fueron procesados con el modelo de detección de poses pre entrenado de TensorFlow.js (mismo que se usa en el frontend); para obtener datos de las coordenadas de puntos específicos del cuerpo en el espacio. Tales datos se guardaron en archivos JSON, que luego conformarían el dataset de entrenamiento.



La generación del dataset de entrenamiento y validación, además de recopilar los JSON de datos, se asistió de data augmentation. El proceso de data augmentation consiste en aplicar variaciones aleatorias a los datos reales que se tienen, variaciones que generan datos sintéticos y, pero aun dentro de los confines de datos realísticos; esto permite más diversidad de datos (TensorFlow, 2023). Con dos scripts de Python se generaron los datos sintéticos y se los colocó de manera aleatoria en datasets de entrenamiento y validación. La aplicación de data augmentation a los datos capturados se realizó seleccionando al azar un JSON, y sumando o restando a las coordenadas de todos los puntos un valor aleatorio correspondiente a moverse entre -2.5 y +2.5 cm en el espacio. Esta copia modificada luego se guardaba como un nuevo archivo de datos.

Se entrenó al MLP durante 250 épocas, learning rate de 0.005, batch size de 200, con 2250 muestras de entrenamiento y 250 de validación. Las métricas usadas sobre los sets de entrenamiento y validación fueron “loss” (pérdida) y “accuracy” (exactitud); estos indican la suma de los errores cometidos con cada muestra (que tan lejos de lo correcto está el análisis), y la cantidad de muestras correctamente clasificadas, respectivamente. Típicamente, se busca baja pérdida y alta exactitud. A continuación, se muestra la evolución de estas métricas durante el entrenamiento, y su estabilización al finalizar el entrenamiento.

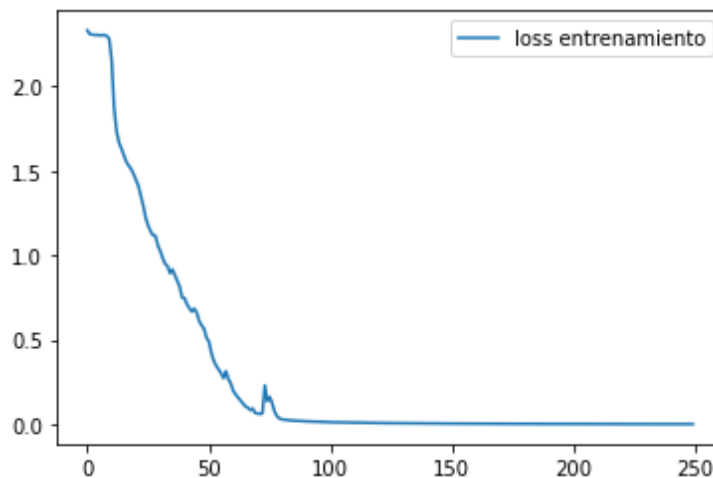


Figura 1 Evolución de pérdida con dataset de entrenamiento

udla

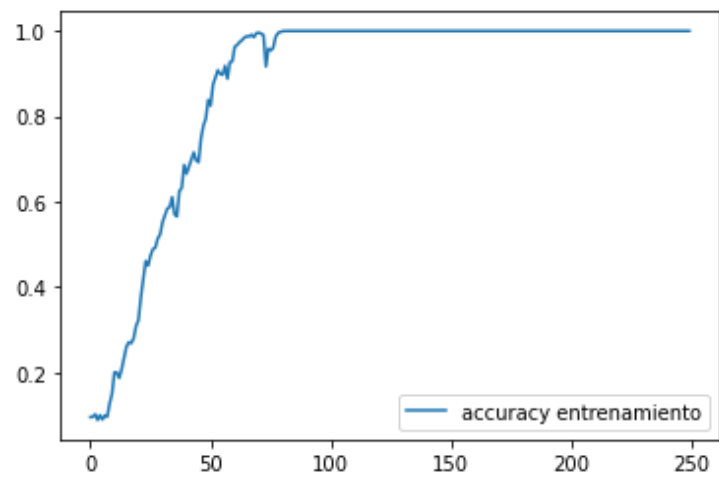


Figura 2 Evolución de exactitud con dataset de entrenamiento

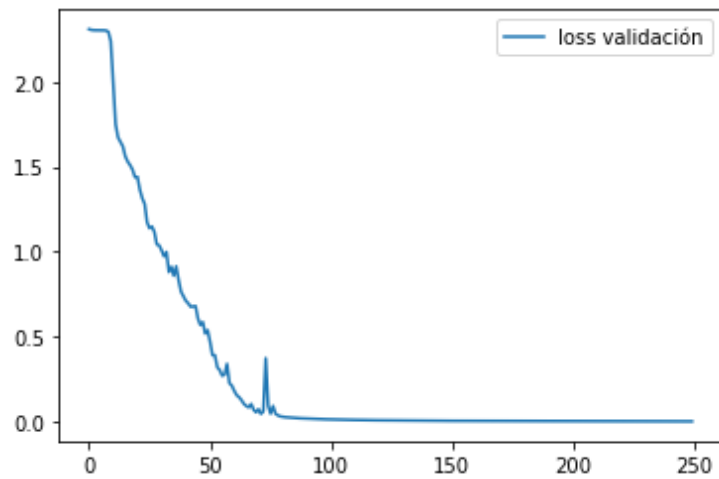


Figura 3 Evolución de pérdida con datos de validación

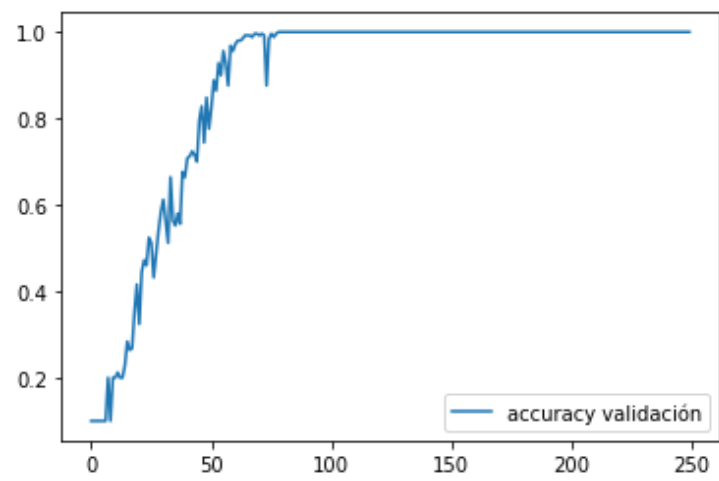


Figura 4 Evolución de exactitud con datos de validación



C. Máquina de puntaje y temporizador

Para crear una máquina (es decir, hardware y software) que sea capaz de mostrar los datos de puntaje, penalizaciones, y tiempo en un combate, se utilizan microcontroladores, luces LED, displays LED, transceptores de radiofrecuencia, y algunos componentes electrónicos. La lógica general de las funcionalidades de la máquina está controlada por un ESP32, un System On a Chip (SoC) que almacena todos los datos mostrados por máquina y también interpreta los comandos recibidos por radiofrecuencia al módulo nRF24L01+ (transceptor de 2.4GHz). A este SoC se le denomina la “unidad lógica” de la máquina, y mediante Universal Asynchronous Receiver / Transmitter (UART, un protocolo de comunicación serial) indica a otros microcontroladores los datos que mostrar.

El control remoto para la máquina consta de 24 botones, que permiten editar el puntaje, tiempo, penalización y prioridad para desempate. El personamiento de estos botones es controlado por la placa de microcontrolador Arduino Mega 2560 Rev3, que a su vez está conectado a un transceptor nRF24L01+. Según el botón presionado, el Arduino Mega utiliza el módulo de radiofrecuencia para enviar el comando correspondiente al receptor de la unidad lógica usando Serial Peripheral Interface (SPI, también un protocolo de comunicación serial). Además de comunicarse en un canal específico, el control remoto tiene un identificador único que debe ser validado por la máquina, permitiendo la posible existencia de múltiples máquinas en un futuro.

La información almacenada y controlada por la unidad lógica es mostrada a los usuarios haciendo uso de luces LED y matrices LED. A las luces se las controla independientemente encendiendo u apagando el circuito de cada una. Por otro lado, las matrices son una combinación de matrices más pequeñas de 8x8 LEDs, controlados por un el chip MAX7219, que recibe datos mediante SPI y controla el encendido de filas y columnas para mostrar datos. Estos elementos de display son controlados por 3 placas de microcontrolador Arduino Nano 3.x. Dos de estos Arduino Nano se conectan a matrices LED de 2x2 para mostrar puntajes, uno controla la matriz 1x4 del temporizador y los LEDs que indican penalizaciones y prioridad. Los tres microcontroladores reciben instrucciones de que datos mostrar con canales independientes de UART desde el ESP32.

Dado que las máquinas de esgrima existentes en la Academia tienen funcionalidades para repetidor, tienen un puerto RJ45 que emite señales eléctricas de 12 voltios para indicar si se dio un toque. Utilizando relays (componentes electromecánicos que cierran un circuito en presencia de corriente) es posible interpretar estas señales. La máquina existe emite 4 señales que indican cual de los esgrimistas hizo un punto (o ambos), y si uno o ambos realizaron toques que no se consideran válidos. Estas señales, al leerse por el ESP32 con ayuda de los relays, pausan el temporizador y, en caso de requerirse según la modalidad de esgrima (es decir, el arma utilizada) se incrementan los puntajes según corresponda. Esta



conexión no es esencial para el funcionamiento del resto de la máquina, pero es una ayuda adicional en los combates para detener el cronometro a tiempo.

Para cumplir con normas de la FIE, y añadir una funcionalidad muy útil a la máquina, también se tiene un circuito de redundancia de poder y persistencia de datos. Un circuito electrónico se encarga de suministrar poder al ESP32 desde baterías en caso de que se pierda la fuente de alimentación principal. La unidad de lógica lee continuamente el voltaje de alimentación para determinar si se ha perdido energía, y de ser así pausa el temporizador y guarda los datos en la EEPROM del sistema. El mismo ESP32 se encarga de también notificar si la batería de respaldo esta baja, mediante un LED. Si se diera el caso de que la redundancia no es necesaria, se puede encender la máquina con o sin conexión a baterías, sin tener que removerlas.

Tanto los microcontroladores (Arduinos) como el ESP32, controlan comunicaciones y electrónica según la lógica de programas hechos en C++. El IDE de Arduino permite crear programas llamados “sketch” en este lenguaje, los cuales se compilan y cargan por USB a los microcontroladores y SoC usando funcionalidades del IDE. Para el control de las matrices LED y comunicación con los transceptores de radiofrecuencia, se utilizan librerías open source disponibles en el repositorio de librerías de Arduino. La librería para los chips MAX7219 de las matrices se encargan de encender los LEDs adecuados con patrones de la librería. La comunicación con los nRF24L01+ se maneja con una librería que se convierte las instrucciones del sketch a comandos aceptados por el chip del transceptor.

La máquina y el control remoto se ensamblaron soldando los componentes en una placa perforada, de modo que permanezcan en esta permanentemente. Para facilitar el ensamblaje, y reemplazo microcontroladores, luces, o matrices LED, todos estos se conectan a sockets (conexiones hembra) soldadas en la placa. Este tipo de conexión asegura una fijación adecuada de los componentes, pero pueden ser removidos sin un proceso largo de remover las sueldas y colocarlas de nuevo. La máquina está contenida en una caja plástica de conexiones eléctricas, ideal también para alojar componentes electrónicos, con perforaciones para botones e indicadores. La tapa de la caja se puede remover para cambiar las baterías de respaldo. El control remoto está ensamblado en una caja de acrílico cortada con láser, con los orificios de acceso a botones requeridos. La tapa posterior del control se puede retirar para reemplazar baterías.

uol/a.

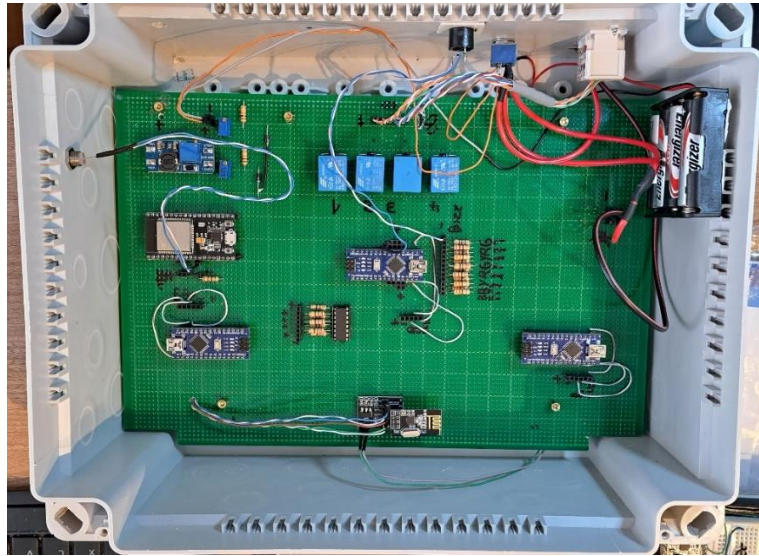


Imagen 6 Conexiones en versión final de la máquina

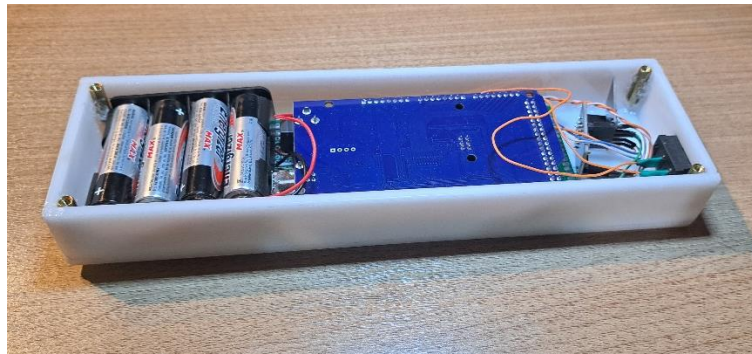


Imagen 7 Conexiones de versión final de control remoto

Una vista simplificada de las conexiones de la máquina de puede visualizar a continuación, si se desea observar la esquemática de conexión completa, se puede hacer referencia al [repositorio del proyecto](#) o al [Anexo 5](#).

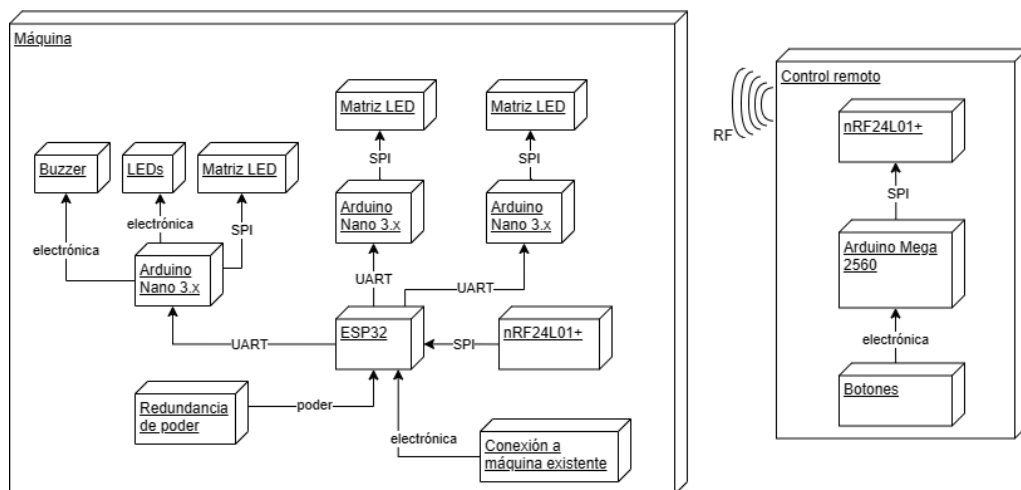


Diagrama 28 Representación simplificada de conexiones de la máquina



El proceso de uso de la máquina se da de la siguiente manera. Con la máquina colocada en un lugar predefinido según la organización de la Academia, esta debe ser encendida, el control remoto también debe encenderse. Después de 1.5 segundos (necesarios para sincronización de microcontroladores y SoC) la máquina se termina de inicializar, lo cual se indica con un pitido. En este estado, el árbitro puede utilizar el control remoto para la edición de puntos, tiempo, asignación de tarjetas de penalidad, y prioridad para desempate. Si la máquina se encendió con la opción redundancia, en cualquier momento cuando se dé una pérdida de poder, se almacenará el estado de la máquina (mientras exista carga en las baterías). Este proceso se representa a continuación.

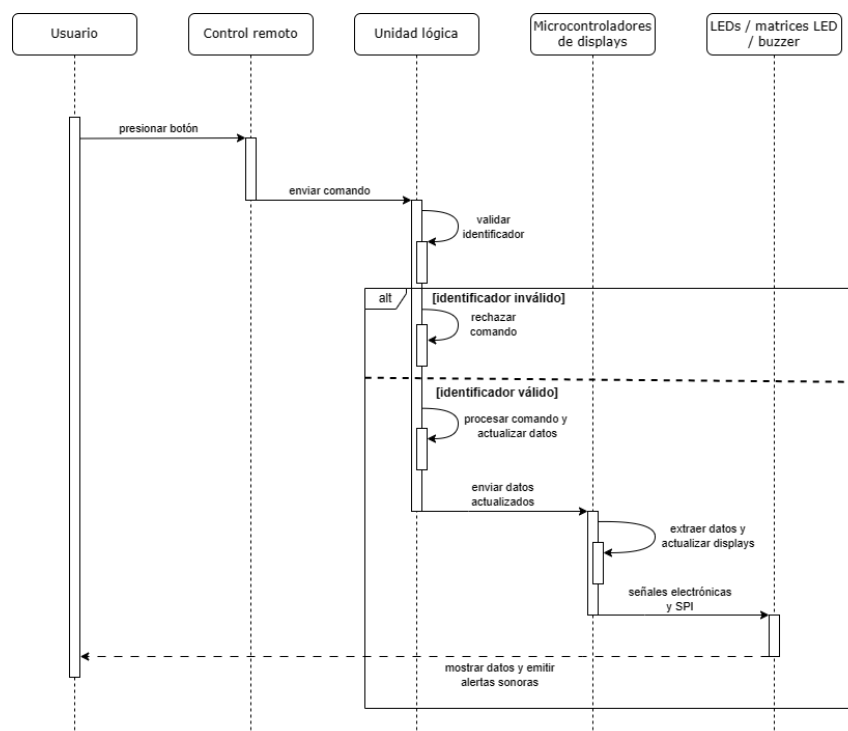


Diagrama 29 Secuencia de uso de la máquina de puntaje

Todo el prototipado de la máquina de puntaje se realizó con los mismos tipos y modelos de componentes que la versión final, pero colocando todo en protoboards o placas de prueba. Una protoboard es una tabla con orificios eléctricamente conectados en filas y columnas, ideal para hacer prototipos pues no requiere soldar los componentes, pero los mantiene fijamente en su lugar (PCMag, s.f.). A una protoboard se conectaron los componentes de la unidad lógica, microcontroladores LEDs y matrices, así como la conexión de datos y poder de las matrices LED. A una protoboard separada se conectaron los componentes del control remoto. De esta manera se pudieron cargar los programas necesarios para pruebas, cambiar y probar componentes para desarrollar la máquina según las funcionalidades establecidas.

uol/a.

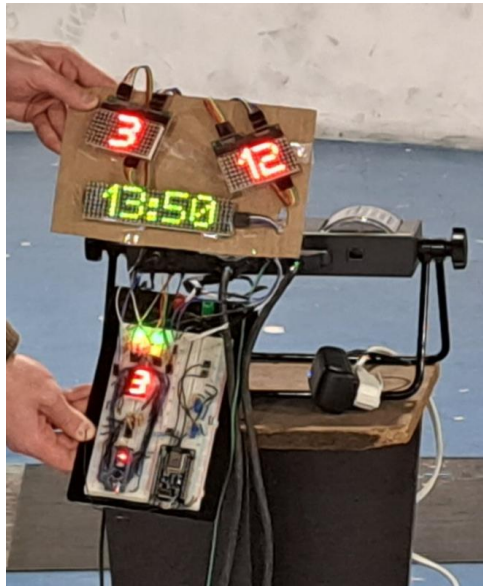


Imagen 8 Primer prototipo de la máquina con un display más pequeño y sin redundancia de poder

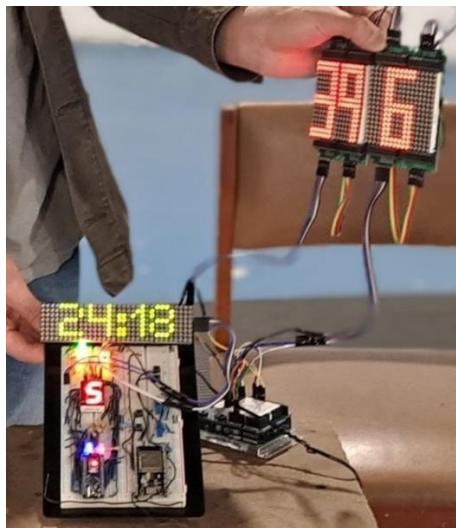


Imagen 9 Segundo prototipo de la máquina con mejoras de displays, aún sin redundancia de poder

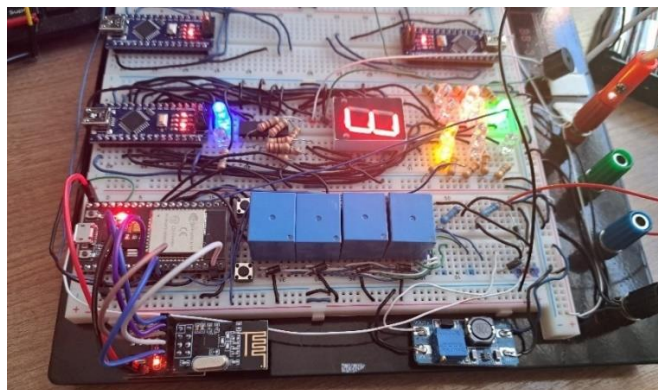


Imagen 10 Prototipo final, con redundancia de poder, receptor de radiofrecuencia, todos los microcontroladores y SoC, y conexión a la máquina existente

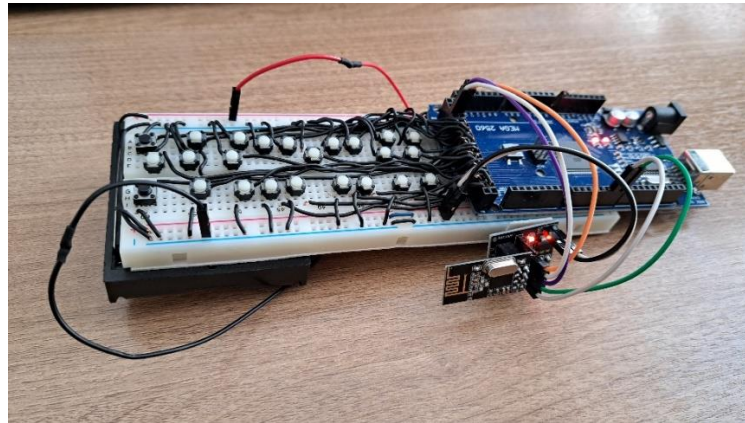


Imagen 11 Prototipo del control remoto, con transmisor, botones, y microcontrolador

7.2.3. Despliegue

Para desplegar la solución en producción se requiere de una plataforma para alojar el sistema web; y ensamblar los componentes de la máquina con su software para que se puedan alojar y usar físicamente en la Academia. Siguiendo lo considerado en el [análisis de costos de la solución](#), para el despliegue de la aplicación web se usa un servidor virtualizado de Digital Ocean. La máquina debe desplegarse ensamblada en las instalaciones de la Academia, para lo cual se requiere un espacio de cerca de un metro cuadrado y conexión eléctrica para alimentación.

Digital Ocean ofrece un servicio llamado Droplets, que consiste en servidores virtualizados con distintas combinaciones de poder de procesamiento, memoria y almacenamiento. Para la presente solución se optó por un servidor con 1 núcleo, 2 GB de memoria y 50 Gb de almacenamiento; lo cual representa un costo de \$12 mensuales. Debido a los recursos necesitados, el frontend, backend y base de datos se ejecutan en el mismo servidor. La máquina virtual corre Ubuntu Server 22.04 LTS, la versión más estable y de soporte prolongado de este sistema operativo. El frontend de la aplicación se sirve directamente desde Apache HTTP Server. El backend, al ser una aplicación de Express.js, se ejecuta como un proceso en segundo plano usando la herramienta PM2 para su gestión; el servidor Apache realiza un proxy a este proceso para exponer el servicio al internet.

El frontend de la aplicación web se sirve al público bajo la URL aecq.fencing-flow.com, el dominio se adquirió con GoDaddy y el certificado SSL para seguridad con Namecheap. GoDaddy permite comprar un dominio por planes anuales, con opciones para servicios adicionales que en este escenario no se usan. El certificado SSL para que la aplicación web cuente con HTTPS se gestionó con Namecheap, quienes proveen certificados de Comodo SSL. Para la solución se utilizan dos certificados, uno para la URL del frontend y otro para la del backend. Los certificados están orientados a páginas pequeñas, y requieren solo demostrar que se es dueño del dominio.



Como se mencionó previamente, la máquina de puntaje se aloja ensamblada en las instalaciones de la Academia. Para ser utilizada se la coloca en un trípode incluido con la misma, y se conecta a un adaptador de 5 voltios DC. Colocándose alado de la máquina esgrima de la Academia, se puede controlar la máquina con el control remoto y conectarla a los tomacorrientes existentes. En caso de quererse conectar la máquina existente, se requiere un cable de red directo (RJ45) para conectar ambas máquinas.

7.3. Pruebas y evaluación de la solución

7.3.1. Proceso de pruebas continuas

De manera paralela al desarrollo del proyecto, se deben evaluar los resultados de cada fase de su elaboración. Es necesario que, con cada producto o MVP resultante las distintas fases de desarrollo y sprints, se tenga una evaluación adecuada. Para esto se siguieron los siguientes criterios:

- Realizar pruebas unitarias y de integración para cada nuevo feature desarrollado en un sprint.
- Cualquier cambio o desarrollo del código de la app web debe ser evaluado por ESLint para asegurar que se sigan buenas prácticas con JavaScript y Node.js.
- Cada desarrollador es responsable de su código, siguiendo la repartición de módulos explicada en la *sección 7.1.3*. En caso de encontrarse problemas con el código que tengan gran impacto en la solución (es decir, que afecten funcionalidades o causasen retrasos mayores), estos deben evaluarse de manera conjunta con el equipo.

Las evaluaciones de los desarrollos realizados deben estar orientadas al cumplimiento de objetivos del proyecto y a asegurar el cumplimiento de funcionalidades propuestas. Además, se deben tener en cuenta estos objetivos de calidad para las evaluaciones:

- Generar una solución que cumpla los requerimientos recopilados del cliente.
- Código mantenible.
- Cumplimiento de buenas prácticas.
- Seguridad de datos de los usuarios.
- Mantener el menor número de defectos en producción para generar software confiable.
- Pruebas continuas a lo largo del desarrollo, no al final de este.

7.3.2. Validaciones con usuarios

Durante el desarrollo de la solución, se mantuvo contacto constante con el cliente para validar el cumplimiento de requisitos. Esto significó reuniones periódicas para validar el diseño (a nivel de interfaz) de la página web y el flujo de los procesos a



realizarse para las funcionalidades. La revisión de características y funciones de la máquina se realizaron con los prototipos, estos eran validados con los entrenadores para definir mejoras o correcciones. Las validaciones llevadas a cabo con los usuarios se pueden clasificar en aprobaciones de diseño, aprobaciones de flujo de funcionalidad, y pruebas de aceptación

A. Diseño

1) Aplicación web

Tras recopilar los requisitos para la aplicación web, estos se plasmaron inicialmente en forma de wireframes que detallaban la estructura de cómo se verían las interfaces del sistema. Esto servía para que los usuarios pudieran validar no solo la forma en que está construida la página, si no también los datos que se podrían mostrar y guardar de una manera más visual que con descripciones de requerimientos. Con este proceso de revisión y aprobación fue posible identificar fallas de interpretación en los requisitos y datos faltantes. Este proceso también fue de gran ayuda para identificar mejoras futuras al sistema, que están fuera del alcance del presente proyecto, pero se pueden implementar como proyectos futuros.

2) Máquina de puntaje

En lugar de presentar esquemáticas y gráficas las validaciones se hacían directamente con los prototipos. No todas las funcionalidades estaban implementadas aún, pero se aseguraba que sea posible visualizar el tamaño, forma y visualización general de los datos que mantiene la máquina. Durante este proceso, dos principales correcciones se pudieron hacer a la máquina. Primero, se incrementó el tamaño de los displays para puntaje, pasando de 2 matrices horizontales (1x2) a 4 matrices en forma cuadrada (2x2). Además, se añadió un LED a los indicadores de tarjetas amarillas y rojas, ahora usándose 2 LEDs para cada tarjeta.

B. Flujo de funcionalidades

1) Aplicación web

Usando como apoyo los wireframes aprobados previamente, se validaban con los usuarios el flujo de las funcionalidades acorde a lo diseñado por el equipo. En este punto se definían las funcionalidades que se podrían implementar como planificado, y aquellas que debían ser revisadas tras recibir retroalimentación. Cualquier funcionalidad que requería correcciones, era revisada en sesiones posteriores. El flujo general de la inteligencia artificial se realizó con el apoyo de una implementación inicial de la detección de poses, esto para determinar tamaños de la imagen de detección y la visualización del esqueleto y puntos detectados por el modelo.



2) Máquina de puntaje

Una vez implementadas las funcionalidades en el prototipo, la validación de estas consiste en confirmar que cumplan los procesos regulares para un combate y normas necesarias de esgrima. Las siguientes validaciones se realizaron:

- Al asignar la segunda (o más) tarjeta amarilla, se convierte en roja y se aumenta un punto al esgrimista contrario.
- Toda tarjeta roja aumenta un punto al esgrimista contrario
- Una tarjeta amarilla ser asignada por primera se convierte en roja si ya se había asignado una roja previamente.
- Durante los últimos diez segundos del temporizador, se muestran decimas de segundo si la máquina esta contando y hasta céntimas cuando se pausa.
- La prioridad (para desempates) se puede asignar manual o aleatoriamente.
- Se pueden editar los minutos, segundos, decimas y centésimas del temporizador.
- El periodo se puede aumentar y vuelve a 1 tras superar 9.
- Los puntos se pueden aumentar y disminuir manualmente.

C. Pruebas de aceptación

1) Aplicación web

Estas pruebas se realizaron con el sistema desplegado en el servidor, de manera que los usuarios puedan ejecutar las funciones del sistema y probarlas. Durante esta fase de pruebas los usuarios pudieron encontrar bugs en la aplicación web, validar funcionalidades, y definir mejoras visuales. Tras las pruebas realizadas, los usuarios confirmaron que las funcionalidades propuestas se habían cumplido en su completitud. Se encontraron algunos bugs en la aplicación web, afortunadamente estos no se trataban de fallas en las funcionalidades, si no problemas solo de ciertos requests HTTP que se pudieron resolver rápidamente para que los formularios funcionen de manera confiable. Estética y visualmente, se describió que el diseño del sitio es adecuado.

2) Máquina de puntaje

Luego de validar todas las funcionalidades y cumplimiento de reglas del deporte en el prototipo de la máquina, se hicieron pruebas de aceptación con la máquina ensamblada. Los entrenadores, cumpliendo en este caso un rol de arbitro, probaron en la Academia la versión final de la máquina ensamblada. Este proceso les permitió validar que se cumplan las funcionalidades de control de tiempo, puntajes, penalizaciones, y prioridad para desempates. El tamaño, luminosidad, y ubicación de los displays en la versión final de la máquina se consideraron correctos.

7.3.3. Integración continua

El proyecto consta con dos archivos de configuración correspondiente a los pipelines de CI (Continuous integration). Estos pipelines ejecutan 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. Tanto el proyecto de frontend como el backend poseen un pipeline individual que se ajusta a la manera de verificar que el sistema esté funcionando correctamente.

El pipeline del frontend realiza los siguientes pasos:

1. Instalar dependencias del proyecto
2. Ejecutar el linter configurado
3. Ejecutar las pruebas unitarias y de integración creadas
4. Crear un build del proyecto

El pipeline del backend realiza pasos similares:

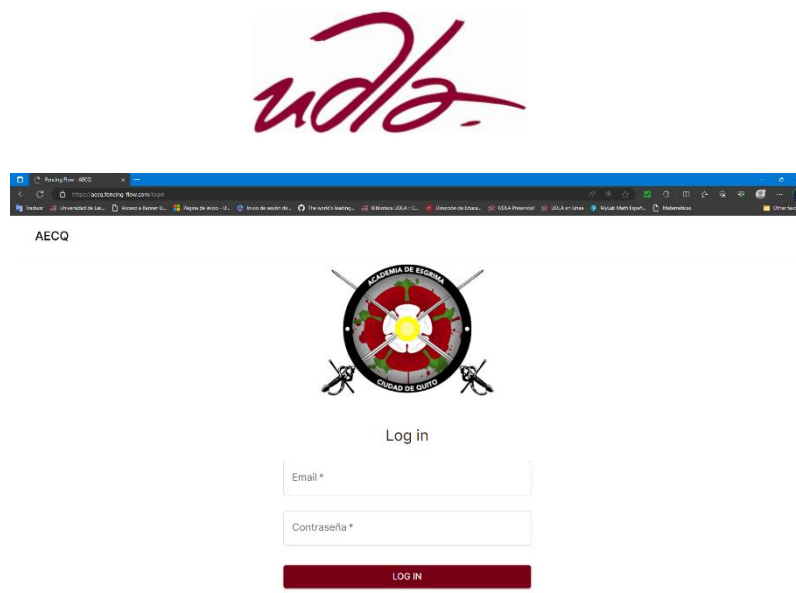
1. Instalar dependencias del proyecto
2. Ejecutar el linter configurado
3. Ejecutar las pruebas unitarias y de integración creadas

La ejecución de estos pipelines se realiza automáticamente al crear un nuevo Pull Request en Git gracias a GitHub Actions, con el fin de verificar que los cambios que se estén proponiendo no afecten la funcionalidad del sistema. Y en el caso de que algún pipeline falle, permite a los desarrolladores visualizar las fallas en el código para poder crear un nuevo cambio que solucione el problema.

7.4. Resultados y Discusión

La implementación de la solución planteada para el proyecto, según las pruebas realizadas para validarlo, cumple con los objetivos y alcance planteados. El desarrollo de la aplicación web resultó en un sitio que es accesible por internet, desde navegadores web para gestionar los datos de entrenamiento, visualizar planes, gestionar planificaciones y retroalimentaciones. La inteligencia artificial para asistencia a entrenamientos individuales es utilizable desde la aplicación web, y sus resultados pueden ser consultados para análisis de entrenadores o esgrimistas. Se pudo diseñar y ensamblar una máquina de puntaje y temporizador, que permite a un entrenador mantener registro de puntajes, tiempo, penalizaciones y usar funciones para desempate.

La aplicación es accesible desde la URL <https://aecq.fencing-flow.com>, la cual es accesible por los esgrimistas, entrenadores y administradores. Tras completar el proceso adecuado de registro, pueden acceder a sus cuentas para utilizar las funcionalidades desarrolladas. Esta web app tiene un diseño responsive, lo que permite utilizarla en una mayor variedad de dispositivos, ya sean computadores de escritorio, laptops o móviles. Esta aplicación web permite la gestión de planes de entrenamiento, retroalimentaciones, combates, esgrimistas y la adecuada visualización de estos datos.

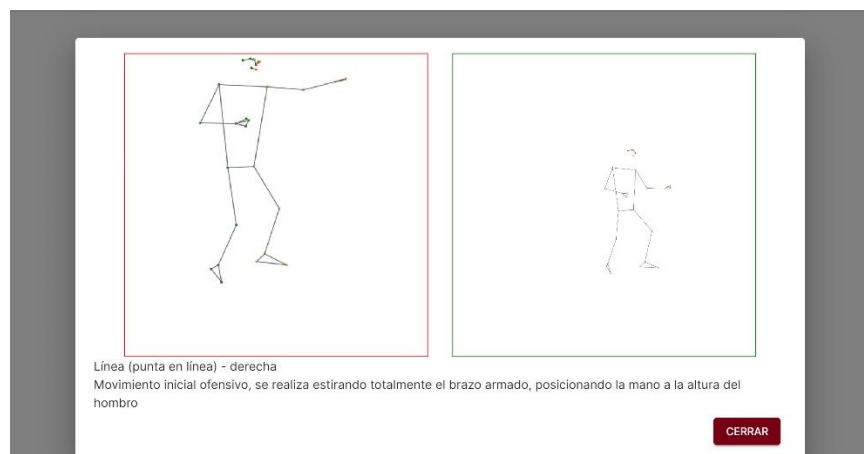


Captura 1 Pantalla de inicio de sesión de la web app



Captura 2 Pantalla de planificación de entrenamiento

Como parte de la aplicación web se integró el módulo de inteligencia artificial para asistir en entrenamientos individuales. La inteligencia artificial permite detectar posibles errores esencialmente mediante descarte. Al capturar el movimiento de un esgrimista, la inteligencia artificial determina si el mismo corresponde a un movimiento correcto, en caso de no serlo, se clasifica como un error y se notifica al usuario. Los errores detectados se guardan para su posterior análisis.



Captura 3 Pantalla de visualización de potenciales errores detectados con inteligencia artificial



La máquina de puntaje y temporizador permite a un entrenador mantener conteo de los puntajes en un combate de manera mucho más eficiente que usando su memoria. Permitiendo también asignar penalizaciones, editar el temporizador, y asistir con resolución de empates mediante prioridad predefinida (según reglas de esgrima), es una herramienta adecuada para facilitar el arbitraje en un combate. Además, la suma del costo de los componentes utilizados está cercano a los \$200 USD, que comparado con máquinas oficiales que oscilan alrededor de \$900 USD, representa una buena alternativa para combates y competencias no oficiales.

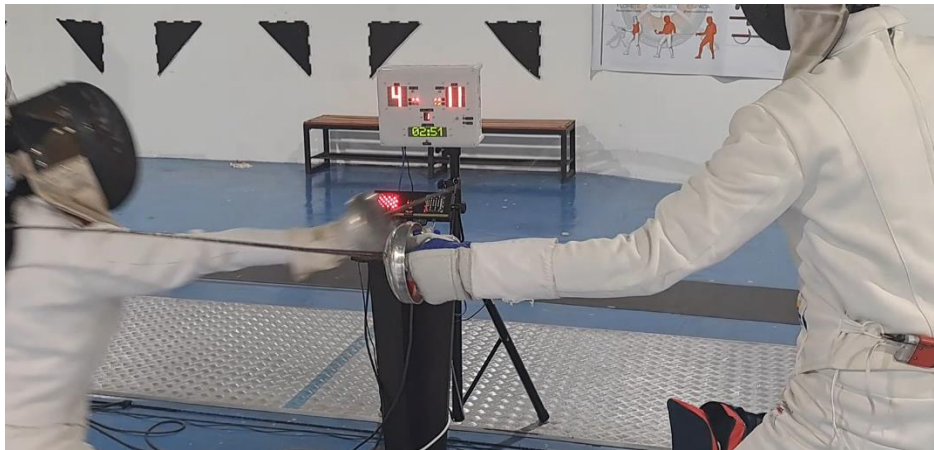


Imagen 12 Máquina de puntaje en funcionamiento durante un combate, conectada al máquina existente con cable de red para pulsos eléctricos

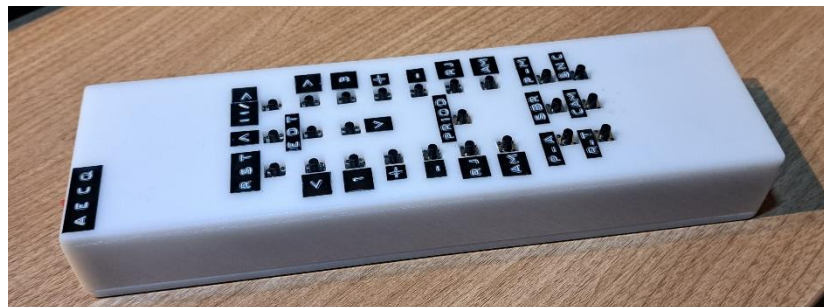


Imagen 13 Control remoto para envío de comandos a la máquina

El sistema web implementado con la inteligencia artificial integrada y la máquina de puntaje han probado ser una solución completa para el problema. Los entrenadores de la Academia están empezando a digitalizar sus procesos, inscribiendo a los esgrimistas para gestionar sus planes de entrenamiento, registrar combates, y visualizar entrenamientos. Gradualmente, se están inscribiendo a más esgrimistas para que consulten sus planes y usen la inteligencia artificial de entrenamiento. La máquina de puntaje y temporizador ha tenido buena aceptación en la Academia, se ha notado que facilita la gestión de datos en un combate. Actualmente el directorio de la Academia se encuentra evaluando el desempeño de la máquina antes de decidir si solicitar que se ensamblen más instancias. Se espera continuar extendiendo las funcionalidades de la solución para atender más necesidades de la Academia.



7.5. Implicaciones éticas

Debido a que la aplicación web maneja datos personales de los esgrimistas para la adecuada gestión de sus actividades en la Academia, es de gran importancia que estos datos sean resguardados. Cumpliendo con los procesos actuales de la Academia solo los administradores y entrenadores tienen el permiso para visualizar los datos de cualquier esgrimista, mientras que los esgrimistas solo pueden visualizar y actualizar sus propios datos. Además, por ello es importante que el sistema valide accesos y permisos de los usuarios y origen (solo el URL del frontend de la página) de todo request de datos a la API.

Es importante que la máquina de puntaje y temporizador cumpla con las reglas definidas para esgrima para permitir combates justos. No se podría predecir que esgrimistas usarán la máquina en qué momento, por lo que no se la puede programar para beneficiar o perjudicar a alguien con antelación. Sin embargo, fallas de lógica de programación que resulten en el incumplimiento de reglas, podrían tener la consecuencia de generar un ambiente injusto para uno de los esgrimistas. Estos fallos podrían también impedir el desarrollo normal de combates, provocando que las actividades de recreación y deporte se vean bloqueadas. Por ello, se ha validado cuidadosamente con los usuarios que se cumplan reglamentaciones y no existan fallos con los efectos adversos mencionados.

8. Conclusiones y Recomendaciones

8.1. Conclusiones

- La implementación de sistemas informáticos, en este caso en la web, para la digitalización de procesos y datos, es un objetivo alcanzable implementándose el correcto análisis de requisitos y diseño de la solución. Ha sido posible crear una aplicación web que permita digitalizar los principales procesos de entrenamiento de la Academia, facilitando la planificación y acceso a datos. A futuro, sería ideal continuar expandiendo las funcionalidades de este sistema.
- El modelo de inteligencia artificial para asistencia en entrenamientos individuales es una herramienta útil para los esgrimistas y entrenadores. Siendo de fácil acceso en un computador, los deportistas tienen más oportunidades para practicar, y sus entrenadores pueden mantener supervisión sobre su desempeño incluso fuera de la Academia.
- Desarrollar hardware asistiendo de microcontroladores y aplicación de electrónica básica ha probado ser una alternativa funcional y más accesible que hardware oficial para esgrima. Puede que la máquina aun necesite mejoras para llegar a un nivel aceptable por la FIE, pero actualmente ha demostrado ser ideal para clubes pequeños, que necesitan una manera confiable de manejar puntajes y tiempo en combates internos.



8.2. Recomendaciones

- Para evitar retrasos por correcciones y rediseños, es ideal un análisis cuidadoso de las entidades del sistema, sus atributos y tipos de datos. En las fases iniciales del proyecto, algunos inconvenientes se presentaron por esta razón. Sería ideal continuar obteniendo experiencia de mejores prácticas en bases de datos relacionales a futuro.
- Una obtención de datos representativos para entrenar un modelo de inteligencia artificial es esencial. Planificación y atención a detalles son necesarias para esto. Aunque es posible asistirse de procesos como feature engineering o data augmentation, siempre es necesario contar con una cantidad adecuada de datos reales lo suficientemente diversos. Sin esto, los datos que se puedan extraer pueden tener parcialidad ante ciertas clases, no ser lo suficientemente diversos para aplicaciones reales en ambientes menos controlados, o ser insuficientes para entrenar un modelo confiable.
- Al trabajar con electrónica, por simplificado que sea gracias a plataformas como Arduino, ESP32, o Raspberry, se requiere de asistencia de multímetro y esquemáticas de conexión. Para evitar daños por malas conexiones, o largas sesiones de diagnóstico de problemas, es recomendable que los prototipos siempre se vean guiados y validados por esquemáticas y mediciones de multímetro. Esto permite ahorrar tiempo y recursos al evitar fallas en funcionamiento o daño al hardware y hasta computadoras usadas para cargar los programas.

9. Trabajo futuro

Se han identificado algunas mejoras a la solución en forma de funcionalidades adicionales que pueden ser el punto de partida para trabajos futuros. Estos trabajos pueden consistir en proyectos independientes o mejoras a añadirse a la solución desarrollada. Los potenciales trabajos futuros identificados se listan a continuación con una descripción para cada uno.

- **Módulo de campeonatos para aplicación web:** casi todos los clubes y academias de esgrima tienen, periódicamente, campeonatos internos o interclubes, en diferentes épocas del año. Para crear un sistema web más completo y que pueda ser potencialmente utilizado por más clubes de esgrima, se pueden añadir funcionalidades para registrar resultados de un campeonato. Tales funcionalidades pueden añadirse a la aplicación web, para aprovechar la existencia de registro de combates y los esgrimistas ya registrados. Para campeonatos entre clubes, también se debe considerar el poder añadir invitados. Dado que algunos campeonatos son pequeños, además de permitir cumplir el proceso y normas de la FIE para campeonatos, se necesitará añadir ciertas flexibilidades que permitan llevar a cabo campeonatos con menos competidores y personal.
- **Conexión a API para máquina de puntaje:** para aprovechar la existencia de una API web, y las funcionalidades WiFi del ESP32 de la máquina de puntaje,



se puede añadir una funcionalidad para cargar los datos de la máquina al sistema web. Configurando el ESP32 para que se conecte a la red WiFi de la Academia, el dispositivo puede enviar un request a la API para almacenar el estado actual de la máquina en la web, y usar estos datos para registrar resultados de combates. Esto convierte a la máquina en un dispositivo IoT.

- **Estudio de comparación de desempeño de inteligencia artificial de captura de poses contra tecnologías de motion capture:** para detección precisa de movimientos típicamente se utiliza tecnología motion capture, que consiste en capturar los movimientos de un sujeto usando marcadores o sensores colocados en el cuerpo (Adobe, s.f.). Esta tecnología es ampliamente usada en cinematografía y ciencias debido a su precisión, pero requiere de un espacio amplio y dedicado, además de ser sumamente costoso. Por otro lado, la inteligencia artificial necesita menos espacio, y puede ser menos costosa de adquirir y mantener. Considerando esto y los avances que sigue presentando la inteligencia artificial, se podría realizar un estudio para comparar a fondo el desempeño de captura de poses y movimientos entre tecnología de motion capture e inteligencias artificiales como la detección de poses de TensorFlow.
- **Aplicación de escritorio dedicada para identificación de movimientos y errores específicos en deportes:** ejecutar constantemente la inteligencia artificial de detección de poses demanda más recursos que una inferencia que se ejecuta solo en ciertos momentos. Debido a esto sería ideal tener una aplicación independiente, que use recursos de un computador (más abundantes que en móviles), y no tenga las limitaciones de ejecutarse en navegador. Además de contar con disponibilidad adicional de recursos, se podría iniciar un proyecto de mayor alcance dedicado enteramente a la extracción de datos, feature engineering (extracción de las características más representativas de un dato), y ajuste de modelos de inteligencia artificial para detección de errores en múltiples deportes, no solo esgrima.
- **Detección de errores específicos en esgrima usando inteligencia artificial en aplicación web:** las capacidades de análisis de la inteligencia artificial en la aplicación web podrían, similarmente al anterior punto, ser aumentadas. Aunque aún dentro de las limitaciones de ejecutarse en un navegador y en Node.js en el servidor, se podría iniciar un nuevo proyecto de extracción de datos (o compartir el usado con el anterior punto, de realizarse) y usarlos para reentrenar el modelo de análisis con más clases (errores puntuales) que detectar. Junto con los ajustes adecuados a nivel de lógica de negocio, esta es otra mejora para un sistema web más completo.
- **Miniaturización y cajas personalizadas para máquina de puntajes:** la máquina de puntaje (y el control remoto) pueden ser montadas en PCB (placa de circuito impreso). Las PCB se pueden diseñar a partir de las esquemáticas de conexión existentes, lo que permitiría soldar los componentes más próximos entre sí, miniaturizando la máquina. Para más miniaturización y mejor acceso



a baterías, sería ideal diseñar modelos 3D de las cajas para la máquina y control remoto para ser impresas en 3D con más personalización.

- **Conexión a eléctrica a esgrimistas para la máquina:** dándole esencialmente todas las funcionalidades de una máquina de esgrima oficial, con algunos cambios de componentes y nuevas conexiones eléctricas para los cables de los trajes de los deportistas, se pueden detectar los toques en un combate. Con una revisión adecuada de diseño y sistema de reloj para cumplir las normas de la FIE de tiempos en los cuales los toques son registrados, el hardware se podría convertir en una alternativa adecuada para máquinas de esgrima, posiblemente siendo certificada por la FIE. Esto reemplazaría la funcionalidad de conexión a máquinas existentes con relays.

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>

Bento, C. (21 de Septiembre de 2021). *Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis*. Obtenido de Towards Data Science: <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>

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

FIE. (2022a). *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

FIE. (Noviembre de 2022b). *Material rules December 2022*. Obtenido de FIE International Fencing Federation: <https://static.fie.org/uploads/31/155253-book%20m%20ang.pdf>

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>
- Microsoft. (06 de Agosto de 2015). *ASP.NET - Single-Page Applications: Build Modern, Responsive Web Apps with ASP.NET*. Obtenido de Microsoft Learn:
<https://learn.microsoft.com/en-us/archive/msdn-magazine/2013/november/asp-net-single-page-applications-build-modern-responsive-web-apps-with-asp-net>
- 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/>
- PCMag. (s.f.). *Definition of breadboard*. Obtenido de PCMag:
<https://www.pcmag.com/encyclopedia/term/breadboard>
- TensorFlow. (07 de 2023). *Data augmentation*. Obtenido de TensorFlow Core:
https://www.tensorflow.org/tutorials/images/data_augmentation
- 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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.



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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas



3. Detección de posibles errores

3.1. Dataset de entrenamiento para la AI

3.1.1. 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 semanas.

3.1.2. 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 evidencias 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 semanas.

3.2. Modelo de AI para reconocimiento de posibles errores

3.2.1. 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 semanas.

3.2.2. 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 semanas.

3.2.3. 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 semanas.

3.2.4. 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 semanas.

3.3. AI de reconocimiento integrada en la app web

3.3.1. 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 semanas.

3.3.2. 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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.

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 semanas.



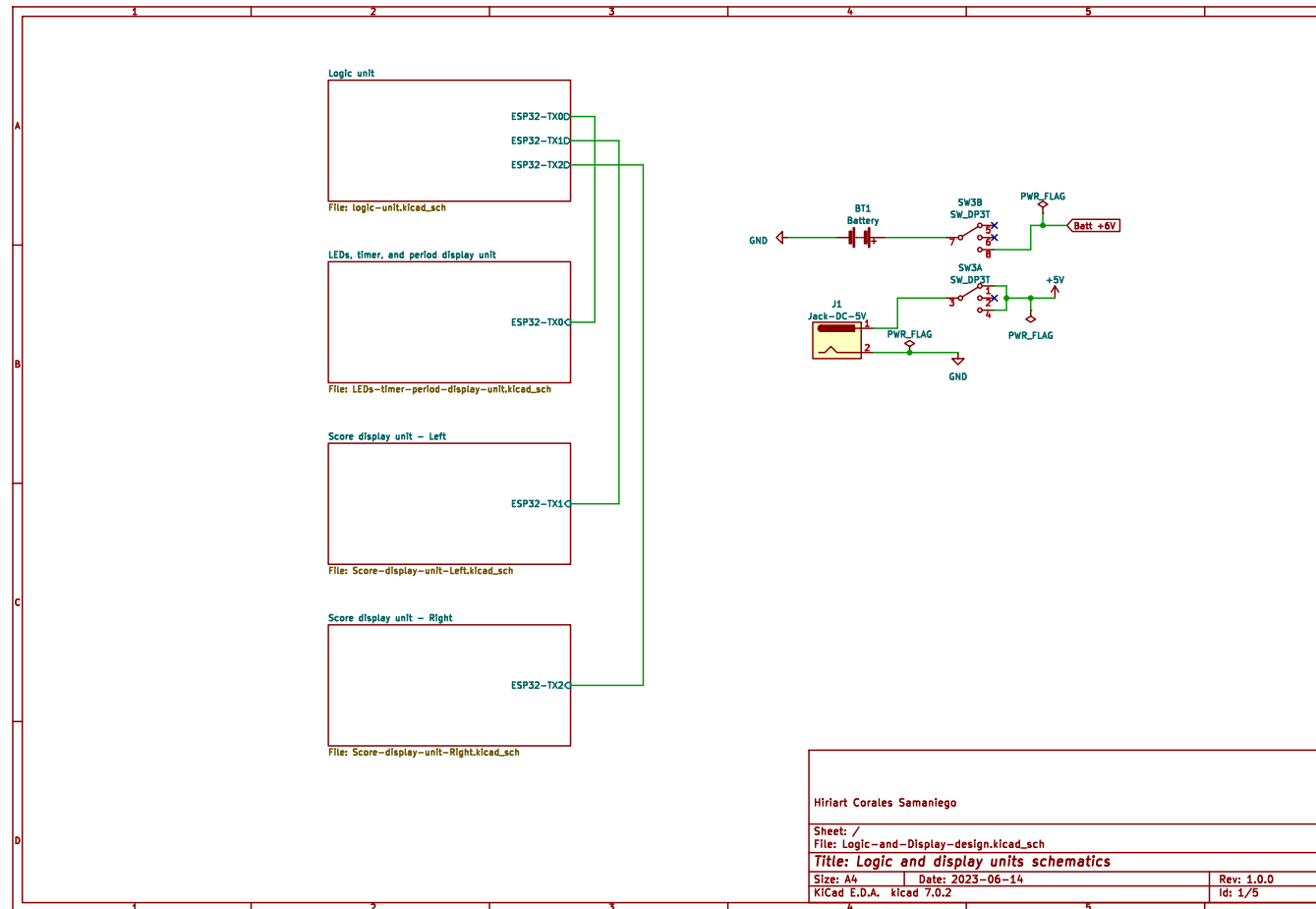
11.4. Anexo 4: planificación en Microsoft Project

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

ud/a.

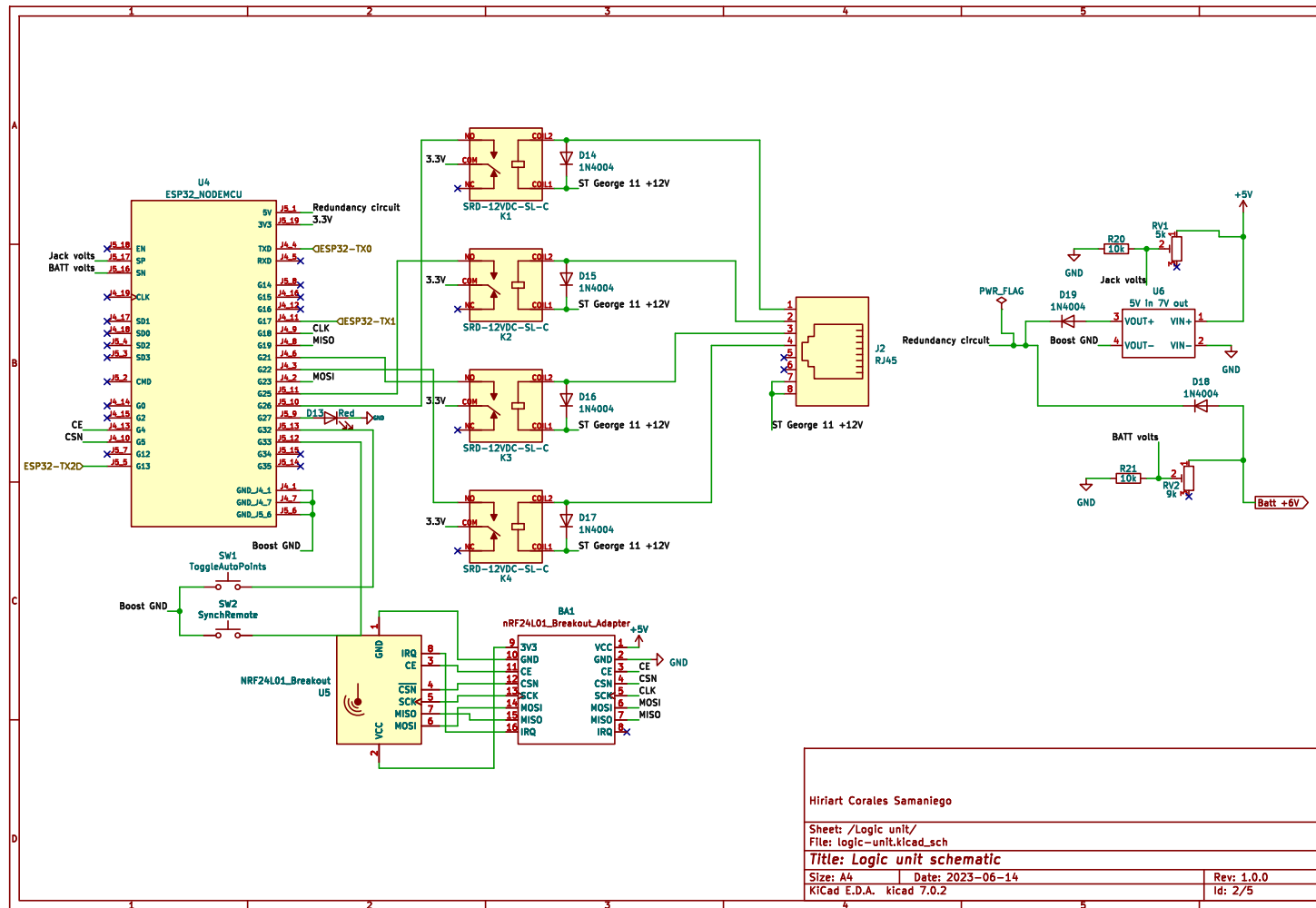
11.5. Anexo 5: esquemática de conexión de la máquina

11.5.1. Unidad de lógica y displays



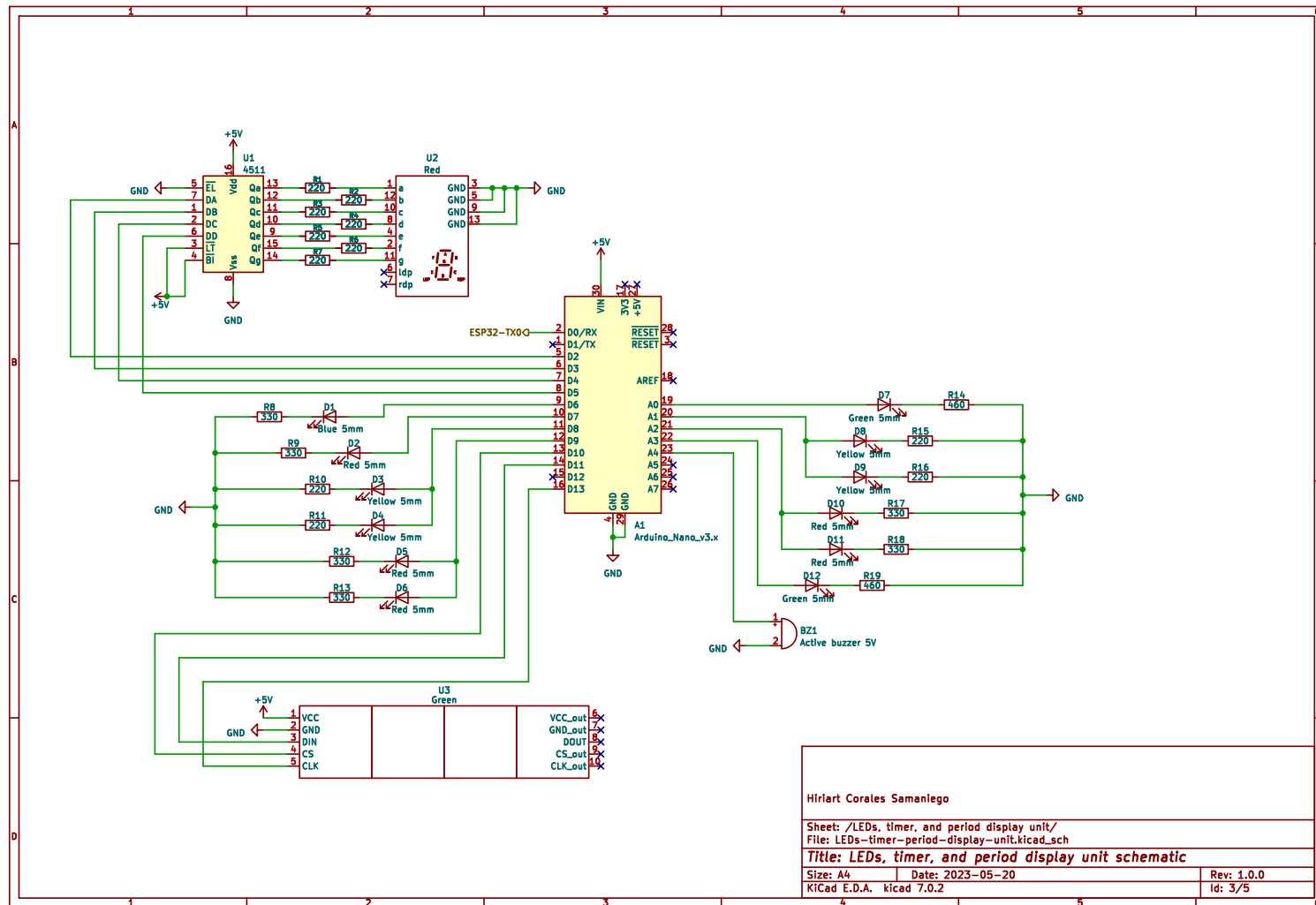
Esquemática 1 Vista general de la máquina de puntaje

uol/a



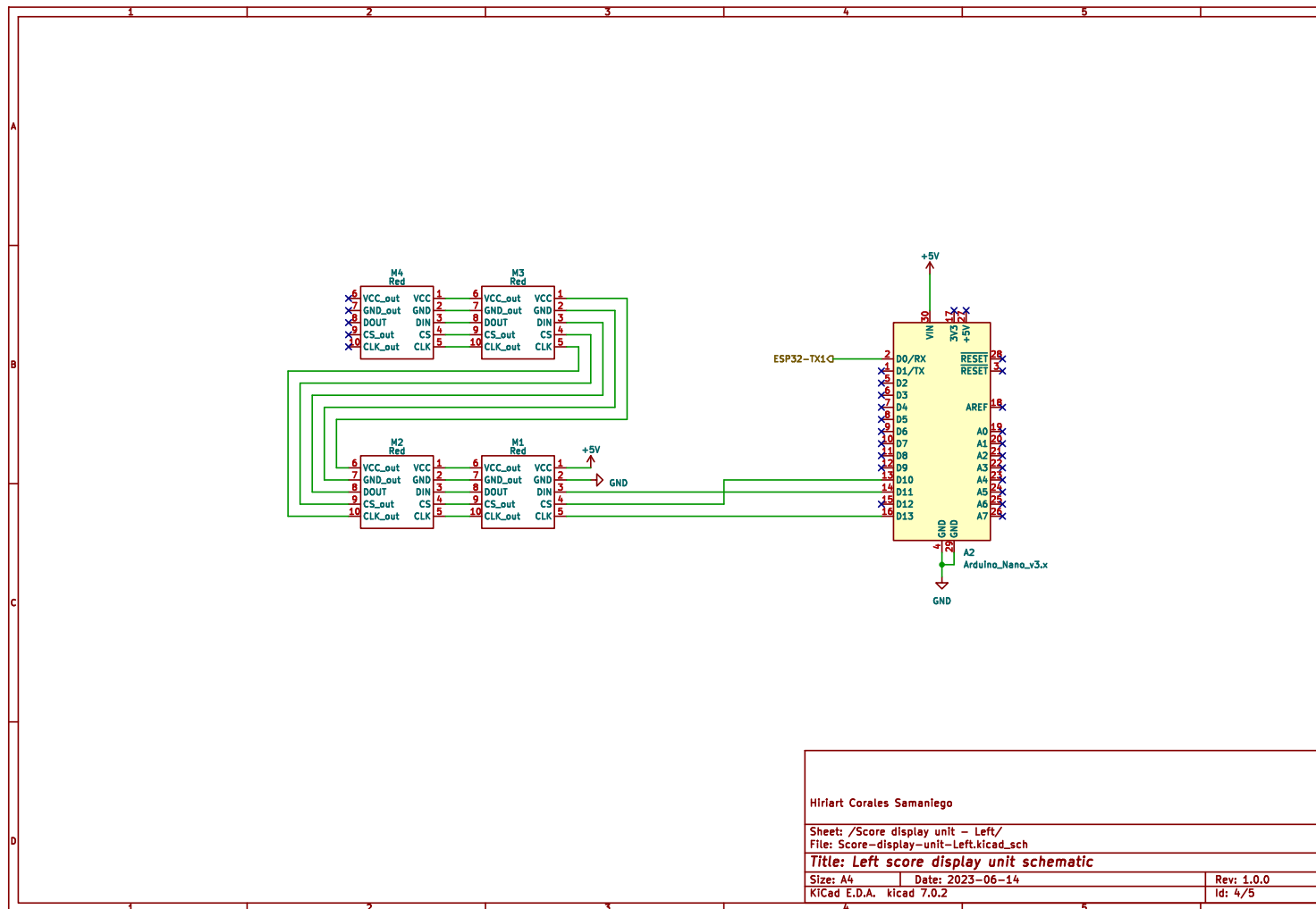
Esquemática 2 Unidad lógica (ESP32) de la máquina, para control de datos, comunicación, y lógica

ucl/a.



Esquemática 3 Arduino Nano para control de displays e indicadores de tiempo, periodo, penalizaciones, prioridad, y alertas sonoras

uol/a.



Esquemática 4 Arduino Uno para control de display de puntos de esgrimista de lado izquierdo

[illegible]

82

