



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

<Tecnología Específica>

TRABAJO FIN DE GRADO

Plantilla de TFG para ESI-UCLM
Curso de \LaTeX esencial

Jesús Salido Tercero

<mes, año>



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

<Primera línea Depto. Director>

<Segunda línea Depto. Director>

<Tecnología Específica>

TRABAJO FIN DE GRADO

Plantilla de TFG para ESI-UCLM
Curso de \LaTeX esencial

Autor(a): Jesús Salido Tercero

Director(a): <director (nombre apellidos)>

Director(a): <codirector (nombre apellidos)>

<mes, año>

Plantilla TFG

© Jesús Salido Tercero, <año>

Este documento se distribuye con licencia Creative Commons Atribución Compartir Igual 4.0. El texto completo de la licencia puede obtenerse en <https://creativecommons.org/licenses/by-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.



TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario: _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

*A mis alumnos ...
(para siempre)*

Resumen

(... versión del resumen en español ...)

El resumen debe ocupar como máximo una página y en dicho espacio proporcionará información crucial sobre el *‘qué’* (problemática que trata de resolver el TFG), el *‘cómo’* (metodología para llegar a los resultados) y los objetivos alcanzados.

Abstract

(... english version of the abstract ...)

Versión del resumen en inglés. En los trabajos cuyo idioma principal sea el inglés, el orden de Resumen y Abstract se invertirá.

AGRADECIMIENTOS

Aunque es un apartado opcional, haremos bueno el refrán «*Es de bien nacidos, ser agradecidos*» si empleamos este espacio es un medio para agradecer a todos los que, de un modo u otro, han hecho posible que el TFG «llegue a buen puerto». Esta sección es ideal para agradecer a familiares, directores, profesores, compañeros, amigos, etc.

Estos agradecimientos pueden ser tan personales como se desee e incluir anécdotas y chascarrillos, pero nunca deberían ocupar más de una página.

Jesús Salido Tercero

ÍNDICE GENERAL

| | |
|---|--------------|
| Índice de figuras | XIX |
| Índice de tablas | XXI |
| Índice de listados | XXIII |
| Índice de algoritmos | XXV |
| 1. Introducción y objetivos | 1 |
| 1.1. Resumen | 1 |
| 1.2. Objetivos | 1 |
| 1.3. Estructura del trabajo | 2 |
| 2. Test de viabilidad | 3 |
| 2.1. Test de Slagel | 3 |
| 2.1.1. Características de Plausibilidad | 5 |
| 2.1.2. Características de justificación | 6 |
| 2.1.3. Características de adaptación | 7 |
| 2.1.4. Características de éxito | 8 |
| 2.1.5. Resultado | 9 |
| 3. Estado del arte | 11 |
| 3.1. Esgrima | 11 |
| 3.1.1. Modalidad de espada en esgrima | 11 |
| 3.1.2. Estructura competición esgrima | 13 |
| 3.2. Sistema de apoyo a la toma de decisión y Machine Learning en esgrima | 14 |
| 3.2.1. Sistema de apoyo a la toma de decisión (DSS) | 14 |
| 3.2.2. Sistemas basados en los conocimientos | 14 |
| 3.2.3. Machine Learning | 15 |
| 3.2.4. Base de datos y Web Scrapping | 16 |

| | |
|--|-----------|
| 4. Propuesta | 17 |
| 4.1. Metodología | 17 |
| 4.2. Extreme Programming | 17 |
| 4.2.1. Valores | 18 |
| 4.2.2. Características | 18 |
| 5. Desarrollo | 21 |
| 5.1. Agile Inception Deck | 21 |
| 5.1.1. ¿Por qué estamos aquí? | 22 |
| 5.1.2. El <i>Elevator pitch</i> | 22 |
| 5.1.3. Diseñar una caja para el producto | 22 |
| 5.1.4. Lista de los no | 22 |
| 5.1.5. Conoce a tus vecinos | 23 |
| 5.1.6. Muestra la solución | 23 |
| 5.1.7. ¿Qué nos quita el sueño por las noches? | 24 |
| 5.1.8. Tamaño del proyecto | 24 |
| 5.1.9. Muestra con claridad lo que se va a dar | 24 |
| 5.1.10. Muestra lo que va a conllevar | 25 |
| 5.2. Iteración 1 | 26 |
| 5.2.1. Adquisición conocimiento básico | 26 |
| 5.2.2. Adquisición conocimiento mediante entrevistas | 26 |
| 5.3. Iteración 2 | 29 |
| 5.4. Iteración 3 | 33 |
| 5.5. Iteración 4 | 35 |
| 5.6. Iteración 5 | 35 |
| 5.6.1. Obtención BBDD | 35 |
| 5.6.2. Tratamiento BBDD | 37 |
| 5.6.3. Extraer conocimiento de los datos | 37 |
| 5.6.4. Usar conocimiento en aplicación | 42 |
| 6. Conclusiones | 45 |
| 6.1. Objetivos alcanzados | 45 |
| 6.2. Trabajo futuro | 46 |
| 6.2.1. Carencias | 46 |
| 6.2.2. Sistema experto | 47 |
| 6.2.3. API predicción | 47 |
| 6.2.4. Aplicación web | 48 |
| 6.2.5. General | 49 |
| A. Test de Slagel | 51 |

| | |
|--|-----------|
| B. Entrevistas | 53 |
| B.1. Entrevista 1 | 53 |
| B.2. Entrevista 2 | 56 |
| C. Diagramas toma decisión | 61 |
| D. Reglas del sistema experto | 65 |
| E. Algoritmo fuerza bruta obtención mejores variables KNN | 73 |
| F. Bocetos herramienta web | 77 |
| G. Algoritmo valores KNN | 81 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| 3.1. Blanco válido espada | 12 |
| 3.2. Dimensiones espada | 12 |
| 4.1. Desarrollo SE | 18 |
| 5.1. Desarrollo SE | 21 |
| 5.2. Desarrollo SE | 23 |
| 5.3. Desarrollo SE | 25 |
| 5.4. Esquema toma decisión | 30 |
| 5.5. Árbol decisión experiencia | 31 |
| 5.6. Arbol decisión distancia con caminos numerados | 31 |
| 5.7. Boceto página llenar | 34 |
| 5.8. Boceto página guiado | 35 |
| 5.9. Chuleta decisión algoritmo ML | 39 |
| 5.10. Mapa conocimiento arriesgar | 43 |
| C.1. Esquema toma decisión | 61 |
| C.2. Árbol decisión experiencia | 62 |
| C.3. Árbol decisión distancia | 62 |
| C.4. Árbol decisión agresividad | 62 |
| C.5. Árbol decisión acortar distancia | 63 |
| C.6. Árbol decisión hierro y segunda intención | 63 |
| F.1. Boceto página inicio | 77 |
| F.2. Boceto página llenar | 77 |
| F.3. Boceto página guiado | 78 |
| F.4. Boceto página resultado | 78 |
| F.5. Boceto página contacto | 78 |
| F.6. Boceto página contacto | 79 |
| F.7. Boceto página sugerencia | 79 |
| F.8. Boceto página entrenamiento | 80 |

ÍNDICE DE TABLAS

| | |
|---|----|
| 2.1. Leyenda | 5 |
| 2.2. Tabla con las características de plausibilidad | 5 |
| 2.3. Tabla con las características de justificación | 6 |
| 2.4. Tabla con las características de adaptación | 7 |
| 2.5. Tabla con las características de éxito | 8 |
| 2.6. Resultados de viabilidad | 9 |
| 3.1. Ejemplo tabla resultados poule | 13 |
| 5.1. Horas del proyecto | 25 |
| 5.2. Costes del proyecto | 25 |
| 5.3. Tabla objeto atributo y valor | 28 |
| 5.4. Tabla objeto atributo y valor | 29 |
| 5.5. Reglas distancia | 32 |
| 5.6. Sujetos prueba prototipo | 33 |
| 5.7. Estructura BBDD Inicial | 36 |
| 5.8. Ejemplo BBDD inicial | 36 |
| 5.9. Estructura BBDD tiradores | 37 |
| 5.10. Ejemplo BBDD tiradores | 37 |
| 5.11. Estructura BBDD final | 38 |
| 5.12. Estructura campo ganador | 38 |
| 5.13. Tabla equivalencias mano | 38 |
| 5.14. Tabla equivalencias mano | 38 |
| 5.15. Estructura BBDD final | 38 |
| 5.16. Parámetros petición predicción | 42 |
| 5.17. | 43 |

ÍNDICE DE LISTADOS

| | |
|------------------------------------|----|
| desarrollodir/altura.clp | 31 |
| anexos/esgrima.clp | 65 |
| anexos/knn.py | 73 |
| anexos/fuerza_bruta.py | 81 |

ÍNDICE DE ALGORITMOS

INTRODUCCIÓN Y OBJETIVOS

1.1. RESUMEN

Pendiente de escribir

1.2. OBJETIVOS

El principal objetivo de este TFG es contribuir al mundo del deporte, en concreto al deporte de la esgrima. En el comienzo de este deporte hay una gran curva de aprendizaje en cuanto al esquema táctico se refiere puesto que al ser un deporte minoritario los recursos que se le dedican son menores por lo que dificulta la expansión de conocimiento y por ende, la adquisición de este mismo tanto a personas que ya lo practican como aquellas que acaban de empezar. Se quiere reducir la inclinación de dicha curva de aprendizaje en el momento que tienes que aprender por ti mismo y necesitas ayuda de los demás para saber que acciones son las correctas y porque están mal tomadas algunas decisiones, al menos, hasta que te puedes valer por ti mismo como tirador que es capaz de identificar las acciones que están ocurriendo y analizar cuales son las mejores decisiones para contrarrestarlas.

Para ello se pretende desarrollar una aplicación la cual sea capaz de llevar a cabo una toma de decisiones con una serie de entradas, las cuales serán aquellas relacionadas con el entorno de un asalto de esgrima, como son las características de los tiradores, como se está desarrollando el asalto, etc.

Esta aplicación será desarrollada llevando a cabo una labor de Ingeniería de Conocimiento junto con una extracción y análisis de datos para describir el problema, extraer el conocimiento de expertos, conceptualizar, formalizar e implementar dicho conocimiento de manera entendible para los usuarios de dicha aplicación. Dicha aplicación será un SBC el cual aglutine todo el conocimiento de los expertos en su conjunto, además del análisis de datos. Dicho sistema tendrá el objetivo darle una respuesta a un tirador de tal forma que este tenga un punto de vista más para tomar sus decisiones, de tal modo que le sea mas fácil alcanzar la victoria. Además de ayudarle a anteponerse a su rival, también servirá como entrenamiento y salir de dudas cuando se quiera mejorar y adquirir conocimiento.

Esto nos lleva a la conclusión de que para alcanzar el objetivo de este TFG tendremos que diseñar un sistema de apoyo a la toma de decisión, con acceso mediante una aplicación web para darle accesibilidad al programa desde cualquier sitio.

El alcance de este proyecto está basado en los recursos disponibles para realizarlo, tanto personas como tiempo. Varios autores han escrito libros para plasmar su conocimiento sobre este deporte, ya sea como plantear la gestión de un club, como preparar a los tiradores para competiciones, como iniciarlos, etc. Este último caso es el de Elain Cheris hablando sobre los fundamentos básicos de la esgrima en las modalidades de florete y espada ya que ambas comparten las bases. Este libro Manual de esgrima consta de 160 páginas en el que se habla sobre el primer año de aprendizaje de una persona que se inicia en el deporte. Para adquirir este conocimiento se requiere de muchas horas de trabajo y

entrevistas con profesionales por lo que automáticamente descartamos la modalidad de sable, ya que hay poco conocimiento reutilizable.

Debido a los motivos expuestos anteriormente la modalidad de sable se dejará para un futuro a modo de ampliación. Respecto a la modalidad de florete es cierto que comparten las bases pero las técnicas específicas y el conocimiento es totalmente distinto, ya que las propias reglas difieren entre ambas modalidades en algunos aspectos, por lo que se podrían utilizar partes del desarrollo pero toda la adquisición de conocimiento, desarrollo del sistema experto habría que realizarlo partiendo de cero. Para ambas modalidades hay que sumar que conseguir expertos resulta de gran dificultad actualmente, cosa que en un futuro lejano, tres años, espero solventar. Todo esto ha llevado a los objetivos expuestos anteriormente.

1.3. ESTRUCTURA DEL TRABAJO

Pendiente de escribir

TEST DE VIABILIDAD

Este capítulo está dedicado a analizar como de viable es el proyecto. De este modo se justificará la realización y continuación del mismo. Debido a que se cuenta con un experto y para realizar el sistema basado en el conocimiento, se llevará a cabo el estudio de viabilidad mediante el Test de Slagel.

2.1. TEST DE SLAGEL

Este test está pensado para Sistemas Expertos. Para la evaluación utilizará una serie de valores para las características los cuales estarán ponderados de una manera específica en función de las mismas. Dicho estudio de viabilidad se divide en tres etapas:

- Definición de características
- Asignación de pesos a cada una de las características
- Evaluación de cada aplicación candidata.

Las características están divididas en cuatro dimensiones:

1. **Plausibilidad:** En esta dimensión se intentará determinar si se cuenta con los medios necesarios desde la perspectiva de la Ingeniería del Conocimiento. Para ello se analizan dos aspectos:
 - **Características del experto:** se evaluará su fama dentro del sector. Esto quiere decir si los demás profesionales del entorno reconocen su trabajo. También se valorará la capacidad de cooperar. Asimismo, otro aspecto a tener en cuenta será si es competente articulando sus métodos y procedimientos de trabajo. Otra cuestión de suma importancia será que el experto se haya enfrentado al problema con anterioridad y que este lo haya resuelto con éxito.
 - **Características de la tarea que lleva a cabo el experto** se analizará el grado de dificultad que tiene esta. Además se valorará si está adecuadamente estructurada y que tipo de habilidades se requieren para su realización.
2. **Justificación:** Se puede dar el caso de que el desarrollo de un Sistema Experto sea posible, pero esto no quiere decir que esté justificado. En esta dimensión se tratará de comprobar la justificación del desarrollo del dicho sistema desde la perspectiva de la Ingeniería del Conocimiento. Para ello se analizará lo siguiente:

- **Necesidad de la experiencia:** se evaluará las características del ambiente donde hay que realizar la tarea. Los principales factores a tener en cuenta será el peligro que hay en el entorno (terreno hostil), además de la escasez de expertos humanos y la necesidad de su presencia en diferentes sitios a la vez.
 - **Inversión a realizar:** se tendrá en cuenta los costes que conllevarán realizar el Sistema Experto frente al retorno de la inversión realizada. También se tendrán en cuenta soluciones alternativas.
 - **Transferencia de conocimiento:** un buen motivo para justificar la realización del sistema experto sería la posible pérdida de conocimiento. En casos como avanzada edad del experto hará que todo el conocimiento adquirido por el mismo pueda perderse.
3. **Adecuación:** se estudiará si el problema es adecuado para ser resuelto con técnicas de Ingeniería del Conocimiento. Algunos problemas podrán ser resueltos mediante algoritmos convencionales o aquellos problemas que requieran de sentido común. Se tendrá en cuenta la naturaleza, complejidad y tipo de tarea.
 4. **Éxito:** se intentarán determinar las posibilidades de éxito del sistema a desarrollar. Para ello se tendrán en cuenta como de entrenadas estén las personas implicadas, que el Sistema Experto tenga una ubicación idónea, que este sea aceptado por los usuarios como una herramienta de mejora y que este en concordancia con sus soluciones junto a la de los expertos.

Se establecerá una categoría de aplicación sobre cada una de estas dimensiones, mediante las cuales se identificará quien es el destinatario de la tarea. Se pueden distinguir tres actores: experto, usuario o directivo, o la propia tarea. Dentro de cada tarea tendremos dos tipos: esenciales o deseables. Las primeras no podrán tener una puntuación mínima de 7 puesto que como su nombre indica, son esenciales. El sistema de puntuación será entre 0 y 10 dependiendo de la importancia relativa de la misma.

Teniendo en cuenta lo explicado anteriormente, el proceso de evaluación del proyecto es el siguiente:

1. Asignar un valor a cada una de las características en cada dimensión. Dicho valor estará entre 0 y 10, cuya escala representará como de presente está respectivamente, siendo 10 totalmente presente. En el caso de que una característica esencial obtenga un valor menor de 7 la aplicación quedará automáticamente descartada.
2. Ponderar el valor de las características respecto a su peso.
3. Multiplicar para cada dimensión los valores ponderados obtenidos anteriormente.
4. Obtener la media para cada dimensión de los valores ponderados de las características. Para ello se calculará la raíz n -ésima del producto obtenido del apartado anterior. Hay que emplear como índice el valor máximo de los índices usados en cada dimensión.
5. Dividir la suma del resultado de cada dimensión entre cuatro (4) pudiendo obtener como máximo un valor de 76,21.

A continuación se detalla el cálculo de la viabilidad del TFG haciendo uso del Test de Slagel. Para mayor comprensión consultar la leyenda de la tabla 2.1.

Tabla 2.1: Leyenda

| Acrónimo | Significado | Rango |
|------------|--|----------|
| CAT | Categoría | |
| EX | Experto(s) | |
| TA | Tarea | |
| IDEN. CAR. | Identificador de la característica | |
| Pi | Identificador de la dimensión de Plausibilidad | P1...P10 |
| Ji | Identificador de la dimensión de Justificación | J1...J7 |
| Ai | Identificador de la dimensión de Adecuación | A1...A12 |
| Ei | Identificador de la dimensión de Éxito | E1...E17 |
| E | Esencial | 0...10 |
| D | Deseable | 0...10 |

Tabla 2.2: Tabla con las características de plausibilidad

| Cat. | Iden | Peso | Valor | Denominación | Tipo |
|------|------|------|-------|---|------|
| EX | P1 | 10 | 10 | Existen Expertos | E |
| EX | P2 | 10 | 9 | El experto asignado es genuino | E |
| EX | P3 | 8 | 9 | El experto es cooperativo | D |
| EX | P4 | 7 | 8 | El experto es capaz de articular sus métodos pero no categoriza | D |
| TA | P5 | 10 | 9 | Existen suficientes casos de prueba; normales, típicos, ejemplares, correosos, etc | E |
| TA | P6 | 10 | 9 | La tarea está bien estructurada y se entiende | D |
| TA | P7 | 10 | 9 | Solo requiere habilidad cognoscitiva (no pericia física) | D |
| TA | P8 | 9 | 8 | No se precisan resultados óptimos sino sólo satisfactorios, sin comprometer el proyecto | D |
| TA | P9 | 9 | 7 | La tarea no requiere sentido común | D |
| DU | P10 | 7 | 9 | Los directivos están verdaderamente comprometidos con el proyecto | D |

2.1.1. Características de Plausibilidad

Fundamentos de plausibilidad

A continuación se fundamentan algunos de los valores elegidos para las características de Plausibilidad

- P1:** Actualmente se dispone de muchos expertos en el sector. Toda sala de esgrima tiene un maestro el cual es un experto, con mayor o menor experiencia, el cual transmite sus conocimientos adquiridos con los años y los sucesos que vivió a sus alumnos. Por lo tanto podríamos decir que hay al menos un experto por sala de esgrima.
- P3:** El experto escogido tiene especial interés en el proyecto, puesto que serviría de gran ayuda para sus alumnos en competencias dado que actualmente es el único en poder dar apoyo a estos en esas situaciones.
- P7:** Unicamente se requiere el conocimiento suficiente y experiencia en competición para poder identificar las acciones del rival para poder decidir que acciones llevar a cabo de manera que se contrarresten las del rival.
- P9:** Al ser una serie de casos con unas entradas y salidas bien definidas, no requiere de un gran ingenio poder llevar a cabo la decisión, una vez tengamos todos los casos, o el mayor número de estos posibles, identificados.

2.1.2. Características de justificación

Tabla 2.3: Tabla con las características de justificación

| Cat. | Iden | Peso | Valor | Denominación | Tipo |
|------|------|------|-------|---|------|
| EX | J1 | 10 | 9 | El experto no esta disponible | E |
| EX | J2 | 10 | 8 | Hay escasez de experiencia humana | D |
| TA | J3 | 8 | 9 | Existe la necesidad de experiencia simultánea en muchos lugares | D |
| TA | J4 | 10 | 7 | Necesidad de experiencia en entornos hostiles, penosos y/o poco gratificantes | D |
| TA | J5 | 8 | 9 | No existen soluciones alternativas admisibles | E |
| DU | J6 | 10 | 9 | Se espera una alta tasa de recuperación de la inversión | D |
| DU | J7 | 10 | 9 | Resuelve una tarea útil y necesaria | E |

Fundamentos de justificación A continuación se fundamentan algunos de los valores elegidos para las características de justificación

- J1:** En competiciones, sobre todo regionales y clubes pequeños, el experto no suele estar disponible puesto que en la mayoría de las ocasiones tiene otras labores como directorio técnico o incluso ser el mismo un participante mas de la competición. En el mejor de los casos de que no tenga ninguna de estas labores lo normal será que tenga a varios alumnos que atender a la vez, por lo que será una situación común que no esté libre.
- J3:** Se puede dar el caso de que dos alumnos de un mismo maestro tengan un asalto en el mismo instante. Este no podrá estar en ambos sitios a la vez y tampoco es aconsejable estar poco tiempo en uno, después ir al otro y así sucesivamente, por lo que se ve la necesidad de este conocimiento en el mismo instante en distintos lugares.
- J7:** Al resolver la tarea de las incertidumbres sobre que hacer en cada una de las situaciones será mas accesible el deporte para aquellos que estén empezando, puesto que no generará esos sentimientos de frustración por no saber que hacer.

2.1.3. Características de adaptación

Tabla 2.4: Tabla con las características de adaptación

| Cat. | Iden | Peso | Valor | Denominación | Tipo |
|------|------|------|-------|---|------|
| EX | A1 | 5 | 8 | La experiencia del experto está poco organizada | D |
| TA | A2 | 6 | 9 | Tiene valor práctico | D |
| TA | A3 | 7 | 9 | Es una tarea más táctica que estratégica | D |
| TA | A4 | 7 | 10 | La tarea da soluciones que sirvan de necesidades a largo plazo | E |
| TA | A5 | 5 | 8 | La tarea no es demasiado fácil, pero es de conocimiento intensivo, tanto propio del dominio, como de manipulación de la información | D |
| TA | A6 | 6 | 9 | Es de tamaño manejable, y/o es posible un enfoque gradual y/o, una descomposición en subtareas independientes | D |
| EX | A7 | 7 | 9 | La transferencia de experiencia entre humanos es factible (experto a aprendiz) | E |
| TA | A8 | 6 | 6 | Estaba identificada como un problema en el área y los efectos de la introducción de un SE pueden planificarse | D |
| TA | A9 | 9 | 8 | No requiere respuestas en tiempo real “Inmediato” | E |
| TA | A10 | 9 | 8 | La tarea no requiere investigación básica | E |
| TA | A11 | 5 | 8 | El experto usa básicamente razonamiento simbólico que implica factores subjetivos | D |
| TA | A12 | 5 | 8 | Es esencialmente de tipo heurístico | D |

Fundamentos de adaptación A continuación se fundamentan algunos de los valores elegidos para las características de adaptación

- A1:** Actualmente el experto no tiene ningún sistema en el que se pueda consultar su experiencia, no hay nada documentado por lo tanto no hay organización alguna.
- A4:** En este caso el sistema no solo sirve para ayudar en el instante que se consulta, si no que también sirve para transmitir dicho conocimiento al deportista, logrando así una mayor comunidad con conocimiento básico sobre el deporte. De este modo con el paso del tiempo será mas fácil que el conocimiento se pueda expandir
- A6:** Puesto que los ataques pueden ser compuestos, se podrán hacer enfoques graduales en los que se lleven a cabos pensamientos y acciones con mayor profundidad, pudiendo dar estos lugar a acciones mas complejas. De igual manera se podrá hacer de una manera mas sencilla en función de las cualidades del tirador.
- A7:** Es algo tan factible como que se lleva haciendo durante mucho tiempo, puesto que son los maestros de esgrima (expertos) quienes pasan su experiencia a sus alumnos a diario en las clases que se imparten.
- A9:** Antes de empezar un asalto de esgrima se ha de tener clara la táctica a seguir, por lo que no serviría de nada reinventarse en mitad del asalto. Por lo tanto se puede llegar a la conclusión de que no es necesaria una respuesta inmediata ya que entre asaltos como mínimo hay un minuto de descanso, tiempo mas que suficiente para obtener una respuesta.
- A11:** Algunas de las características que se comparan entre tiradores son totalmente objetivas, como la altura, pero otras como la experiencia la rapidez y la frialdad serán cosas subjetivas que se han de percibir.

2.1.4. Características de éxito

Tabla 2.5: Tabla con las características de éxito

| Cat. | Iden | Peso | Valor | Denominación | Tipo |
|------|------|------|-------|--|------|
| EX | E1 | 8 | 9 | No se sienten amenazados por el proyecto, son capaces de sentirse intelectualmente unidos al proyecto | D |
| EX | E2 | 6 | 9 | Tienen un brillante historial en la realización de esta tarea. | D |
| EX | E3 | 5 | 6 | Hay acuerdos en lo que constituye una buena solución a la tarea | D |
| EX | E4 | 5 | 8 | La única justificación para dar un paso en la solución es la calidad de la solución final | D |
| EX | E5 | 6 | 9 | No hay un plazo de finalización estricto, ni ningún otro proyecto depende de esta tarea | D |
| TA | E6 | 7 | 10 | No está influenciada por vaivenes políticos | E |
| TA | E7 | 8 | 5 | Existen ya SSEE que resuelvan esa o parecidas tareas | D |
| TA | E8 | 8 | 7 | Hay cambios mínimos en los procedimientos habituales | D |
| TA | E9 | 5 | 9 | Las soluciones son explicables o interactivas | D |
| TA | E10 | 7 | 8 | La tarea es de I+D de carácter práctico, pero no ambas cosas simultáneamente. | E |
| DU | E11 | 6 | 8 | Están mentalizados y tienen expectativas realistas tanto en alcance como en las limitaciones | D |
| DU | E12 | 7 | 9 | No rechazan de plano esta tecnología | E |
| DU | E13 | 6 | 8 | El sistema interactúa inteligente y amistosamente con el usuario | D |
| DU | E14 | 9 | 9 | El sistema es capaz de explicar al usuario su razonamiento | D |
| DU | E15 | 8 | 9 | La inserción del sistema se efectúa sin traumas; es decir, apenas se interfiere en la rutina cotidiana de la empresa | D |
| DU | E16 | 6 | 9 | Están comprometidos durante toda la duración del proyecto, incluso después de su implementación | D |
| DU | E17 | 8 | 8 | Se efectúa una adecuada transferencia tecnológica | E |

Fundamentos de éxito A continuación se fundamentan algunos de los valores elegidos para las características de éxito.

- E1:** La idea de llevar a cabo este proyecto fue totalmente respaldada por el experto una vez que se comentó, involucrandose y formando parte de él desde el primer momento.
- E9:** Todas las soluciones se pueden explicar argumentando los motivos que da el experto por las que fueron tomadas, de tal manera que el usuario sea capaz de entenderlas.

2.1.5. Resultado

En la siguiente tabla se muestra el resultado de la evaluación de las diferentes dimensiones siguiendo las fórmulas enunciadas en el test de SLAGEL. Una vez evaluadas dichas dimensiones se obtiene la media y se normaliza el valor, es decir, se expresa en tanto por ciento.

Tabla 2.6: Resultados de viabilidad

| Característica | π (Valor total) | π (Peso total) | Resultado | Resultado VC | Resultado máximo |
|----------------|---------------------|--------------------|---------------------------|--------------|------------------|
| Plausibilidad | 3,1752e9 | 2,38085568e9 | $(7,559692955e18)^{1/10}$ | 77.24 | 86.63 |
| Justificación | 3,584e6 | 3,31e6 | $(1,18e13)^{1/7}$ | 73.73 | 85.37 |
| Adecuación | 3,75e9 | 1,03e11 | $(3,87e20)^{1/12}$ | 51.95 | 82.75 |
| Éxito | 9,83e13 | 2,96e15 | $(2,91e29)^{1/17}$ | 54.59 | 81.30 |
| VC Total | | | | 64.38 | |
| VC Normalizado | | | | 84.01 | |

Conclusión

El porcentaje obtenido en la evaluación es suficiente como para seguir adelante con el proyecto, además si normalizamos el porcentaje sube hasta el 84.01 %, porcentaje mas que suficiente para confiar en la viabilidad del proyecto.

ESTADO DEL ARTE

3.1. ESGRIMA

La esgrima es un deporte de estrategia en el cuál tendrás que analizar a tu oponente a la vez que te defiendes de sus acometidas. Al mismo tiempo, tendrás que disfrazar tus ataques con otros para que el oponente no sea capaz de analizar tus movimientos. Es por esto por lo que la mayoría de practicantes lo denominan el ajedrez en movimiento puesto que todo son ataques, por un franco u otro, pequeñas batallas que te llevarán a ganar la guerra al final. Jugar con la mente del rival y calmar la tuya para tener superioridad táctica.

Por supuesto que el físico influye en este deporte, no deja de ser un deporte de contacto en el cual las cualidades físicas (fuerza, agilidad, rapidez, coordinación, reflejos, etc) son un factor mas a tener en cuenta, pero esto no serán mas que componentes de una ecuación la cual nos dará la victoria.

Ahora pasaremos a explicar por encima en que consiste un asalto de esgrima. Un asalto de esgrima es un enfrentamiento entre dos oponentes los cuales tienen que llegar al límite de tocados antes que el rival o haber obtenido mayor puntuación una vez haya terminado el tiempo de asalto. Dependiendo de la modalidad y categoría variaran estos tiempos y límite de tocados. ¿Pero que es un tocado? Un tocado no es mas que un punto a tu favor, el cual se puede conseguir tocando al rival o mediante sanciones del rival. Un ejemplo de sanciones podría ser un comportamiento antideportivo, salirse de la pista por el fondo, dar varias veces la espalda, perder el tiempo repetidas veces mientras vas perdiendo, etc.

Como la mayoría de artes marciales no es mas que un esquema táctico en el cuál tendremos unas variables de entrada mediante las cuales determinaremos una salida. ¿Pero como es posible que un deporte de contacto se base en una serie de entradas y salidas? Bien, un ejemplo muy básico es el siguiente: ante una acción ofensiva directa hacia la parte superior del cuerpo lo lógico es cubrirse esta parte. Aquí es donde entra en juego la estrategia de cada componente del combate. Si el atacante sabe que tu reacción ante una amenaza arriba será cubrirte esa zona, el amagará con un ataque falso (finta) a una parte del cuerpo y sobre tu acción defensiva para evitar esta acometida aprovechará para atacar otra zona que dejaste descubierta por defender la primera acción. Por otro lado, el defensor puede analizar al rival y saber que el primer ataque no será el verdadero, si no que será una preparación para atacar sobre otra zona, de este modo anticiparse y atacar sobre esta preparación o amagar con defenderse sobre la primera zona para después cubrirse la segunda y contra-atacar.

3.1.1. Modalidad de espada en esgrima

Una vez tenemos unas nociones básicas sobre como funciona un esquema táctico en general sobre cualquier disciplina de arte marcial o deporte de contacto, pasemos a hablar de la esgrima en concreto. Hay tres disciplinas dentro de este deporte: sable, florete y espada. Siendo la primera una modalidad en la que se puede tocar con cualquier parte de la hoja, mientras que en las dos últimas son armas de estoque, es decir, solo vale tocar con la punta. Puesto que cada modalidad tiene unas normas y

la espada es la mas practicada y mas sencilla de todas, nos centraremos en ella. En la modalidad de espada se puede tocar en cualquier parte del cuerpo. Con la única excepción de la nuca, puesto que es la única zona en la que no hay protección, para ello hay normas evitando que des la espalda y expongas esta zona tan delicada.

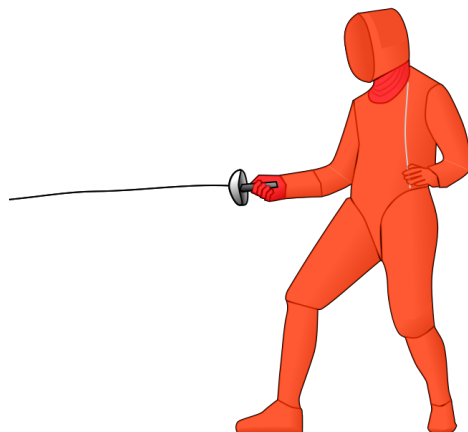


Figura 3.1: Blanco válido espada

Como hemos mencionado antes es un arma de estoque, por lo que el mecanismo de activación estará en la punta y será mediante un botón, el cuál al presionarse sobre un blanco válido (cualquier objeto que no esté aislado en el circuito) cerrará un circuito electrico cuyo objetivo es señalar el tocado. A partir de este momento el rival tiene un breve periodo de tiempo, 0.4 segundos, para realizar un tocado sobre el rival y que haya un tocado doble. Pasado este tiempo el circuito se bloqueará y solo será efectivo el primer tocado. A pesar de que haya un tocado doble no quiere decir que siempre sean válidos ambos tocados. Mediante las normas se dictaminará si los dos, solo uno o ninguno de ellos lo es. Un ejemplo podría ser que uno de los dos tiradores se encuentre fuera de la pista, lo cual anularía su tocado. Como se han podido dar cuenta, hemos hablado de tiempo, por lo que otro componente a tener en cuenta es ser mas rápido que el rival, esto habrá que tenerlo en cuenta en nuestro esquema táctico para poder decidir una acción en la cual, aunque nos toquen, nosotros lo hagamos con suficiente antelación al rival de modo que su tocado no sea válido.

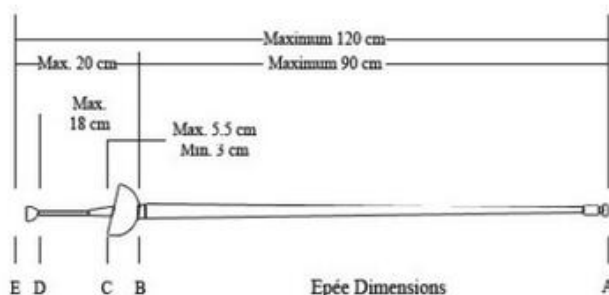


Figura 3.2: Dimensiones espada

Tal y como se habló antes los asaltos tienen un límite de tiempo y un límite de tocados, este será otro factor a tener en cuenta en nuestro esquema táctico sobre como plantear el asalto. Puede que a veces nos interese llevar un asalto hasta el final del tiempo desgastando físicamente al rival para aprovechar esta superioridad al final. Otras veces quizás nos interese lo contrario, acabar con el asalto cuanto antes para evitar dejar al contrario pensar. Puede que otras veces nos interese alargar

el asalto al mayor número de tocados posibles puesto que tengamos mayor repertorio que nuestro oponente, mientras que en el caso contrario, si tenemos pocas acciones nos interesará hacer el menor número de tocados. También habrá que tener en cuenta el marcador y cuanta ventana hay hasta el final del combate, si al rival le falta un tocado para ganar, mientras que a nosotros nos faltan tres, no nos interesa que haya un tocado doble puesto que el ganaría. Estas son algunas de las variables entran dentro de la formula para plantear nuestra táctica en un asalto de esgrima.

3.1.2. Estructura competición esgrima

Una vez que ya sabemos como funciona un asalto de esgrima podemos hablar sobre como funciona una competición de esgrima. Se explicará el funcionamiento de una competición estandar, el cual puede variar en función de la categoría y tipo de competición, puesto que existen varios formatos, como por ejemplo equipos, veteranos, competiciones amistosas, etc. En cuanto a las competiciones individuales primero se disputa una fase de grupos, la cual se denomina *poule* en la cual se dividen a todos los tiradores en poules (grupos) de seis o siete tiradores en función del número de participantes que haya. Siendo siete el número ideal y dejando las de seis en caso de que no haya número suficiente de tiradores. Estas poules se hacen en función del ranking de los tiradores inscritos a la competición, de manera que estén lo mas equilibradas posibles. A destacar que hay un sistema para realizarlas y no es a intuición del directorio. Una vez organizadas los poules se da comienzo a ellas. En ellas se enfrentan todos los tiradores entre ellos, empezando los que sean del mismo país y/o club, para evitar favoritismos mas adelante. Estos enfrentamientos serán en un único asalto con un límite de cinco tocados y una duración máxima de tres minutos. El primero que llegue al límite con diferencia de un tocado o quien tenga mayor puntuación al acabar el tiempo será el ganador de este encuentro. El orden de enfrentamiento entre tiradores también está pre-establecido según la posición dentro de la poule. En los anexos se podrá encontrar una ejemplo de hoja de poule.

Tabla 3.1: Ejemplo tabla resultados poule

| | Tirador 1 | Tirador 2 | Tirador 3 | Tirador 4 | Tirador 5 | Tirador 6 | Tirador 7 |
|-----------|-----------|-----------|-----------|----------------|-----------|-----------|-----------|
| Tirador 1 | x | V | | | | | |
| Tirador 2 | 1 | x | | | | | |
| Tirador 3 | | | x | | | | |
| Tirador 4 | | | | x | | | 2 |
| Tirador 5 | | | | | x | | |
| Tirador 6 | | | | | | x | |
| Tirador 7 | | | | V ₃ | | | x |

La anterior tabla sería un ejemplo de una tabla de poule en mitad de una competición. Se puede apreciar como se anotan las victorias, las derrotas y los resultados de ambas. En caso de obtener una victoria se anotará la puntuación. Una vez terminadas todas las poules se obtendrá la clasificación general, obteniendose de la siguiente manera:

1. Porcentaje Victorias/Derrotas
2. Tocados dados - Tocados recibidos
3. Tocados dados

En caso de empate de todo lo anterior ambos mantendrán el mismo número de serie y se saltará el siguiente. El orden de quien estará encima de otro será aleatorio. Una vez obtenida la clasificación general de las poules, se hace un corte para eliminar a un porcentaje de los participantes, suele ser un veinte por ciento. Con los tiradores restantes de este corte se hace un tablón lo suficientemente grande para acoger a todos los participantes. El número de este tablón será una potencia de dos, es decir 2, 4, 8, 16, 32, 64, etc. En caso de no haber participantes suficientes para completar el tablón los primeros participantes pasarán exentos de la primera ronda. El resto de la competición es una

eliminatória directa en la que el vencedor pasa a la siguiente ronda mientras que el perdedor termina la competición.

Debido a que cada punto cuenta desde el inicio, ya que esto determinará como de fácil será el camino en la competición todas y cada una de las decisiones que tomemos deberán ser lo mas óptimas posibles. Para ello muchas veces contamos con nuestra experiencia, entrenadores o incluso compañeros que nos ayudarán a tomar una decisión dándonos su opinión. Pero como no siempre será este el caso, se quiere desarrollar una herramienta la cual nos facilite la toma de decisiones en mitad de la competición. Para ello usaremos un sistema de apoyo a la decisión, a partir de ahora lo denominaremos como DSS (del inglés Decision Support System).

3.2. SISTEMA DE APOYO A LA TOMA DE DECISIÓN Y MACHINE LEARNING EN ESGRIMA

3.2.1. Sistema de apoyo a la toma de decisión (DSS)

Siempre se ha querido tomar una buena decisión y no siempre se ha podido, en la mayoría de los casos por no tener los conocimientos suficientes. Pero ¿qué es la mejor, una buena y una mala decisión? Esto es algo que, dependiendo del contexto, podría ser tanto subjetivo como objetivo. Por ejemplo, si nuestro objetivo es conseguir correr una maratón, no entrenar para ello posiblemente sea una mala decisión. Sin embargo, si hemos decidido seguir un plan de entrenamiento, el cual nos preparará para terminar nuestro objetivo habremos tomado una buena decisión. Pero esto sigue sin habernos contestado la pregunta de cuál será la mejor decisión. Bien, en este caso tendremos que entrar mas en detalle para saber cual es nuestro objetivo específico, en el caso de que solo sea terminarla, la mejor decisión será seguir aquel plan de entrenamiento que con el menor esfuerzo nos permita terminarla. Sin embargo, si nuestro objetivo es el de conseguir una buena marca, esta no habrá sido la mejor decisión puesto que tendremos que seguir un plan el cual nos permita ir mas rápido, aunque el esfuerzo sea mayor.

Una vez aclarada que es una mala, buena y la mejor decisión podremos hablar sobre lo que es un sistema de apoyo a la toma de decisiones. Estos son sistemas desarrollados para dar una decisión, en su mayoría aplicaciones informáticas, los cuales siguen el proceso de una toma de decisiones. Estos utilizan los datos y modelos para generar las alternativas posibles y acabar tomando la mejor decisión. Detrás de estos sistemas suele haber un sistema experto detrás, el cuál es un conjunto de reglas obtenidas a través de conocimiento extraído previamente. Dicho conocimiento se puede apoyar de otros sistemas como arboles de decisión o redes bayesianas para obtener conocimiento.

Un ejemplo de lo mencionado anteriormente fue en el DSS basado en redes bayesianas con una aplicación a la lucha contra las infecciones nosocomiales (Hela Ltifi, et al). En este caso lo que se pretendía era reducir el número de pacientes infectados. Para ello desarrollaron un sistema para ayudar a los médicos a prevenir dichas infecciones. Dicho sistema fue desarrollado mediante un proceso KDD el cuál se nutre de una base de datos cedida por un hospital. Dicho proceso consistía en analizar los datos e intentar obtener conocimiento de ellos. Para ello habría que seguir una serie de pasos como la limpieza, pre-procesamiento de datos, detección de patrones y ahí es cuando se podría obtener conocimiento. Dicho sistema ayudaría a reducir costes en los hospitales además de reducir el número de casos de infección.

3.2.2. Sistemas basados en los conocimientos

Parte de un DSS pueden ser los sistemas basados en conocimientos, los cuales parten de un conocimiento extraído de un experto y es transportado a una aplicación informática la cual será lo mas

fiel a reproducir la decisión de dicho experto. Para ello se siguen una serie de procesos para poder transformar dicho conocimiento hacia la aplicación. Esta suele contar con una interfaz de usuario para que sea lo mas cómoda posible. El proceso no es trivial puesto que se necesita mínimo de un ingeniero de conocimiento y de un experto en la materia. Además ambos deben estar predispuestos a llevar a cabo el proyecto, puesto que es de vital importancia que se colabore en la mayor medida posible. Para ello se deberá extraer el conocimiento del experto mediante diversos métodos como puede ser un sistema de entrevistas, en las cuales el ingeniero le plantea una serie de problemas y/o dudas al experto y este deberá responderle. Una vez finalizada el ingeniero analizará el resultado de la entrevista, pasando a reglas dicho conocimiento o elaborando una nueva entrevista en caso de que fuera necesario.

Un ejemplo de lo mencionado anteriormente sería el sistema desarrollado en la universidad de Split. Aquí desarrollaron un sistema experto el cual era capaz de detectar talento en jóvenes que practicaban diferentes deportes con sus modalidades. En este caso iban un paso mas allá puesto que no solo jugaban con el conocimiento extraído del experto, si no que generaban una serie de pesos para sus reglas en conjunto de una solución ya hecha en los deportes. Esto llevó a lograr una aplicación en tiempo real mediante una página web en la cual cualquiera podría realizar consultas acerca de si una persona podría ser talentosa. A pesar de todos los esfuerzos en el propio artículo mencionan que no hay una respuesta definitiva y completa a la pregunta.

Por tanto nos encontramos ante la siguiente pregunta ¿Cómo nos podríamos aprovechar los sistemas basados en conocimiento, sistemas de apoyo a la decisión y todo lo mencionado anteriormente en un campeonato de esgrima? Bien pues la respuesta es desarrollar un sistema de apoyo a la toma de decisión, ayudado de un sistema basado en el conocimiento. Esto nos permitirá tener una visión mas en el campo de batalla, donde nuestras capacidades para tomar decisiones están limitadas, ya sea por cansancio físico, presión del momento, etc. Esto no sustituirá a un entrenador, puesto que hay cosas subjetivas y que requieren de un mayor contexto que se le pueda dar al sistema, además de la *intuición* que se tiene en esos momentos pero si nos servirá para tener una visión más de lo que se puede hacer, que en la mayoría de los casos nos hará falta. A pesar de todo esto recordamos que no tomará la decisión por nosotros, sino que nos aconsejará y será el usuario final el que deba tomar la decisión.

3.2.3. Machine Learning

Machine Learning (ML) es una disciplina científica que se encuentra dentro de la Inteligencia Artificial en la cual se crean sistemas que aprenden automáticamente. Entendemos por aprender como la detección de patrones complejos en gran cantidad de datos. En este caso, quien aprenderá será un algoritmo que revisará los datos y será capaz de predecir comportamientos en un futuro.

Dentro del Machine Learning tenemos dos tipos de aprendizajes:

- **Aprendizaje supervisado:** En este tipo tendremos una serie de datos de entrenamiento los cuales habrán sido etiquetados previamente. Con este tipo de aprendizaje tendremos clasificadores automáticos sin tener que programarlos. Podremos elegir la forma que tendrán.
- **Aprendizaje no supervisado:** Este tipo de aprendizaje es el recurrido cuando los datos que tenemos no están etiquetados para el entrenamiento. Es por esta razón por la que se procura encontrar algún patrón que simplifique el análisis.

Usaremos diferentes técnicas de ML para obtener conocimiento de una base de datos, de este modo podremos reforzar nuestro sistema basado en el conocimiento, haciendo de este un sistema mucho mas completo.

3.2.4. Base de datos y Web Scrapping

Ante la inexistencia de una base de datos con datos relevantes para el propósito de este proyecto se decide generar una propia como ya se hizo previamente en el experimento mencionado anteriormente del DSS para los entrenamientos. En su caso los datos fueron generados mediante varios sensores y datos de entrenamientos apuntados por ellos mismos. En nuestro caso generaremos nuestra propia base de datos mediante la obtención de estos mismo a través de internet. Para ello consultaremos las pertinentes páginas y se usará un proceso de automatización de recogida de datos como es Web Scrapping.

Web Scrapping es un proceso de automatización para la recolección de datos de la web. Las páginas más accesibles para ser scrapeadas serán aquellas cuyo lenguaje se ejecute en el cliente como es el caso de HTML o XML, es decir, aquellos en el que se descargue el código fuente y lo interprete el navegador. Para ello primero se descarga el código fuente de la página de la que queremos obtener los datos para después navegar por ella y obtener la información que nos sea necesaria. Además, podremos seguir navegando, visitando otros enlaces de la propia página e interactuar con ella como si fuera un humano quien la visitara desde su computador. Un ejemplo se puede ver en la aplicación que hicieron para un sistema que te ayudara a encontrar trabajo en el cual utilizaban redes bayesianas y scrapeaban los datos que necesitaban para generar la base de datos de la cual pudieron generar las redes bayesianas y después entrenarlas.

PROPUESTA

En este capítulo se hablará sobre como se ha desarrollado el proyecto. Para ello primero hablaremos de la metodología utilizada para después pasara a explicar cuál fue el proceso de desarrollo del sistema.

4.1. METODOLOGÍA

Para poder escoger una buena metodología tenemos que saber de los recursos que disponemos además de cuales son nuestros objetivos (funciones a desarrollar en que tiempo). Previamente se hablo de los objetivos pero un resumen de ellos sería desarrollar una aplicación, con la mayor accesibilidad posible para ayudar al mundo de la esgrima. Para ello se llegó a la conclusión de que se desarrollaría un sistema de apoyo a la decisión el cuál tendría una interfaz web, de este modo se podría acceder a ella desde cualquier sitio con internet. Este proyecto ha de desarrollarse en un periodo de tiempo de unas 300 horas aproximadas por lo que no podremos llegar a desarrollarlo por completo, de modo que este será un factor importante a la hora de elegir la metodología. La que escojamos deberá favorecer el trabajo en funcionalidades completas, de modo que cada vez que se empiece una, se deberá acabar.

Otra cosa a tener en cuenta son los recursos de los que se dispone. En este caso se dispone de un desarrollador el cual hará a su vez de Ingeniero del Conocimiento. También se dispone de un experto en la materia, cuyas horas no serán contabilizadas. Además dichas personas no están a tiempo completo dedicadas al proyecto, solo podrán dedicarle tiempo de forma ocasional. Esto hace que sea mas complicado el desarrollo del mismo, por lo que no se podrán tener varias funcionalidades abiertas sin acabar.

Además sabemos cual es nuestro punto de partida, al igual que nuestra meta, pero el camino es en su mayoría incierto. Esto hará que sea realmente difícil definir una serie de tareas, con pasos a seguir, las cuales tengan una duración estimada fiable.

Por todo lo mencionado anteriormente deberemos escoger una metodología de desarrollo ágil, la cual nos permita adaptarnos a los posibles cambios e inconvenientes que vayan surgiendo en el propio desarrollo. También deberemos escoger una metodología incremental, la cuál nos permitirá aumentar los objetivos de la aplicación en cualquier momento del desarrollo.

4.2. EXTREME PROGRAMMING

La metodología utilizada ha sido eXtreme Programming (XP en adelante). Esto es debido a que es una metodología iterativa-incremental la cual permite el desarrollo de aplicaciones de modo que en cada iteración aumente la funcionalidad de la misma. Una de las principales ventajas que nos ofrece es su cercanía a la improvisación. Como se dijo antes necesitaremos una metodología la cual nos permita improvisar, puesto que desconocemos el camino a seguir. XP también nos facilita el incremento de

funcionalidades con el paso del tiempo, puesto que se centra en sacar versiones estables del proyecto y no pasar a la siguiente hasta que no este estable. En la figura 4.1 se puede ver el flujo normal trabajando con extreme programming

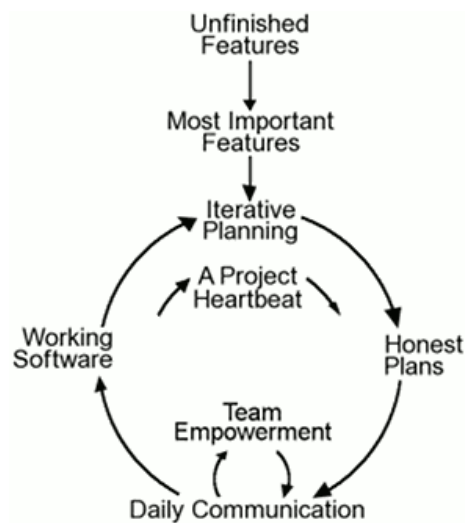


Figura 4.1: Desarrollo SE

4.2.1. Valores

Los valores que se defienden en esta metodología son los siguientes:

- **Simplicidad:** Esta es la base de XP. Cuanto mas simple sea el diseño, mas sencillo será de programarlo. Es por esto por lo que deberemos tener especial atención a las definiciones de los problemas. Para ayudar a un código sencillo se facilitará la refactorización de código cuanto sea necesario en cada iteración. De este modo no generaremos un código muy complejo según añadamos funcionalidades. La documentación del código no debera ser muy extensa ya que un buen código estará autodocumentado y se entenderá por si mismo.
- **Comunicación:** Reforzando el anterior apartado, la comunicación ha de ser sencilla. Evitar los comentarios en el código y mejorar este para que se explique automáticamente es algo fundamental. Por otro lado el desarrollo de test unitarios facilita la comunicación, puesto que estos explicarán de una manera muy clara los casos de uso.
- **Retroalimentación:** En esta metodología el cliente está muy implicado. Los tiempos en los que se hacen desarrollo y se entrega al cliente son muy cortos, de modo que se pueden comprobar funcionalidades poco después de desarrollarlas. De este modo evitaremos en gran medida los desarrollos que después de enseñarselos al cliente se desechan.
- **Valentia:** Para esta metodología tenga éxito hay que ser valiente tomando decisiones para que el diseño no se complique demasiado, puesto que esto conllevaría mayor tiempo de desarrollo, lo cual haría verse comprometida la faceta de entregas en poco tiempo.
- **Respeto:** Todos los miembros del equipo han de ser respetados. Todos los miembros del equipo aportan algo a él y no funcionaría sin ellos. Desde la dirección se respetará a dar responsabilidades a cada uno de los miembros.

4.2.2. Características

Lo que caracteriza a esta metodología es lo siguiente:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas,** frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java, DUnit orientada a Delphi, NUnit para la plataforma.NET o PHPUnit para PHP. Estas tres últimas inspiradas en JUnit, la cual, a su vez, se inspiró en SUnit, el primer framework orientado a realizar tests, realizado para el lenguaje de programación Smalltalk.
- **Programación en parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. La mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- **Integración del equipo de programación con el cliente o usuario.** Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores** antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- **Refactorización del código** reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad en el código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Antes del inicio del desarrollo será necesario ponerse de acuerdo con todos los miembros involucrados en el proyecto. Para ello se utilizará *Agile Inception Deck* el cual estará formado de un conjunto de dinámicas las cuales permitirán ayudar a llevar al proyecto a la dirección deseada. La parte de *Inception* estará formada por diez preguntas, las cuales se detallan a continuación:

1. **¿Por qué estamos aquí?**

Esta cuestión nos permitirá contextualizar el proyecto, respondiendo el motivo del mismo.

2. **El *Elevator pitch***

Literalmente sería el discurso del ascensor. Este apartado intentará dar respuesta a las preguntas qué, por qué y para qué. Para ello se utilizará el símil del tiempo que dura un viaje en el ascensor.

3. **Diseñar una caja para el producto**

Se dará una visión del producto desde la perspectiva del cliente.

4. **Lista de los no**

Aquí detallaremos que no es el producto y que no está dentro de él.

5. **Conoce a tus vecinos**

Daremos a conocer a las personas que están involucradas en el proyecto.

6. Muestra la solución

Se mostrará la metodología y arquitectura que se utilizará en el desarrollo del proyecto.

7. ¿Qué nos quita el sueño por las noches?

En este apartado se intentarán predecir los posibles imprevistos surgidos durante el desarrollo del proyecto.

8. Tamaño del proyecto

Planificación a alto nivel de la duración del proyecto.

9. Muestra con claridad lo que se va a dar

Identificar los temas más importantes en el desarrollo del proyecto

10. Muestra lo que va a conllevar

Análisis de costes del proyecto.

DESARROLLO

En este capítulo se mostrará todo el proceso que se ha seguido para el desarrollo del prototipo de apoyo a la toma de decisiones en esgrima. Para ello se ha seguido la dinámica *Agile Inception* para el inicio del proyecto y después utilizando el desarrollo iterativo e incremental para las primeras versiones.

Para la resolución del proyecto han sido necesarias cinco iteraciones para cubrir los objetivos propuestos en el Capítulo 1 (Ver figura 5.1).

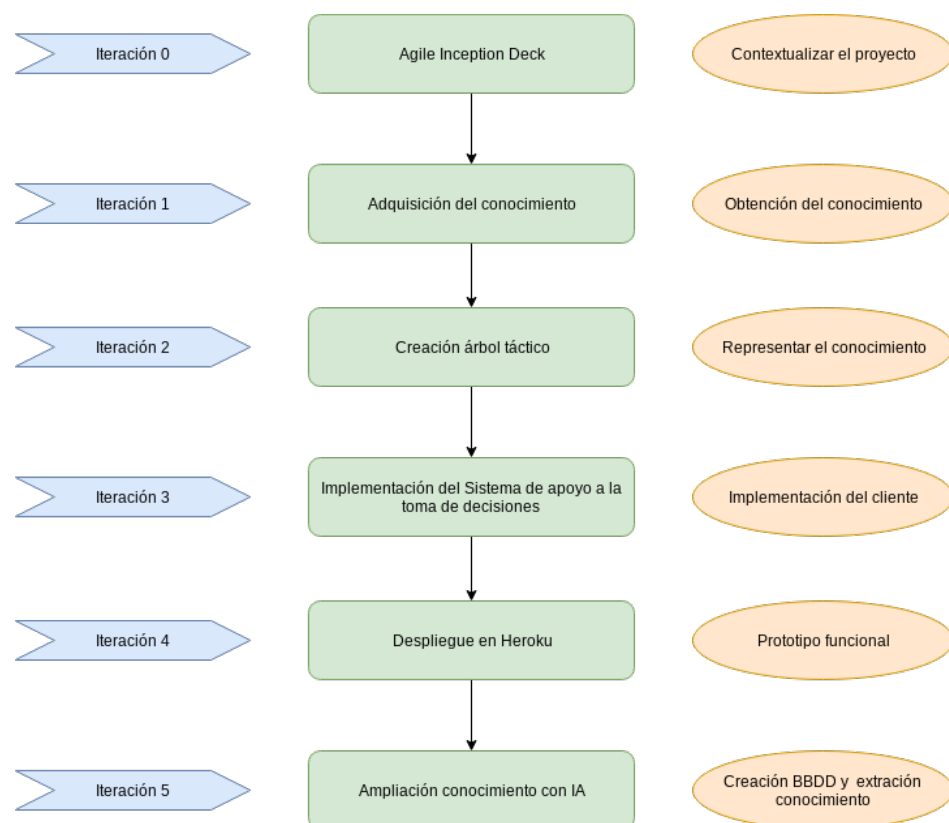


Figura 5.1: Desarrollo SE

5.1. AGILE INCEPTION DECK

Puesto que el proyecto junta dos disciplinas muy diferentes como son Informática y Esgrima, es imprescindible que todas las partes del proyecto que están involucradas remen en la misma dirección. Para ello se utilizará la metodología *Agile Inception Deck*

5.1.1. ¿Por qué estamos aquí?

La necesidad de un apoyo a la toma de decisiones en una competición de esgrima es innegable. En la élite los tiradores tendrán entrenadores dedicados a ellos con estudios previos de rivales. Pero en niveles inferiores no se dará el caso, por lo que el desarrollo de un sistema de apoyo a la toma de decisiones será de gran utilidad para la comunidad de esgrima en las competiciones.

Además, en las salas de esgrima muchas veces no se dispone del tiempo suficiente para que un entrenador pueda transmitir su conocimiento a los alumnos, o no hay un entrenador como tal puesto que algunos clubes han empezado siendo algunos amigos que han practicado el deporte y querían que hubiera un club en su ciudad, sin tener estos muchos conocimientos. Para estos casos en los que se tienen dudas sobre que hacer en ciertas situaciones, este sistema dará una respuesta teórica para aclarar su visión.

Volviendo al mundo de la competición, también servirá a aquellos competidores que quieran averiguar si la elección que hicieron fue la correcta. Después de un torneo son muchos los que piensan que ha pasado, pudiendo usar la herramienta para saber si fue muy disparatada la elección.

5.1.2. El *Elevator pitch*

Aquí se tratará de explicar la finalidad del proyecto con la mayor brevedad posible. Para ello se elaboró la siguiente definición del prototipo:

Sistema de apoyo a la toma de decisión en un asalto de esgrima haciendo uso del conocimiento extraído de expertos y análisis de datos. Para llegar a dicho sistema se utilizarán técnicas de Ingeniería del conocimiento junto a técnicas de Inteligencia Artificial.

5.1.3. Diseñar una caja para el producto

El prototipo del proyecto surge de la necesidad de apoyar (o cubrir en caso de que no exista) la figura de entrenador en un asalto de esgrima para el arma de espada. Suponiendo que el producto final cubra las tres armas la imagen idónea para el producto sería:

- Ayudar a los competidores y entrenadores de esgrima a poder plantear una táctica antes del primer asalto y cambiarla en los descansos teniendo un sistema que le de opciones según las características de los tiradores.
- Ayudar a los practicantes de esgrima para que puedan ampliar y afianzar conocimientos sobre posibles situaciones sin tener que exponerse a ellas.
- Ayudar a la pérdida de practicantes de este deporte en las primeras competiciones haciendo que la curva de aprendizaje sea menor.

El prototipo limita la imagen idónea a un arma, pero la finalidad es la misma para el resto.

5.1.4. Lista de los no

Dado que es un prototipo se han definido una serie de limitaciones para disminuir la duración del proyecto:

- Alcance del prototipo:
 - El prototipo será diseñado únicamente para la categoría de espada.

- El prototipo contemplará las acciones básicas e intermedias de espada.
 - El prototipo contemplará las combinaciones básicas de espada.
 - El prototipo contará con dos versiones. Una de ellas será de uso rápido (preguntas y respuestas sencillas), mientras que la otra será mas detallada en la introducción de sus entradas y en las salidas que de.
- Fuera del alcance:
 - Sistema para las categorías de florete y sable.
 - No estarán contempladas todas las acciones posibles dentro del prototipo.
 - No estarán contempladas todas las combinaciones posibles dentro del prototipo.

5.1.5. Conoce a tus vecinos

En esta sección hablaremos sobre las personas involucradas en el proyecto. Los componentes del proyecto son los siguientes:

- Autor del proyecto:
 - Gregorio Baldomero Patiño Esteo
- Director del proyecto:
 - Dr. José Ángel Olivas Varela.
- Expertos implicados:
 - Juan Lomas Rayego
- Clubs implicados:
 - Espadas de Calatrava (Ciudad Real)

5.1.6. Muestra la solución

En este apartado se muestra un resumen de la arquitectura del proyecto a alto nivel (Ver figura 5.2) y las herramientas empleadas para el desarrollo del mismo.

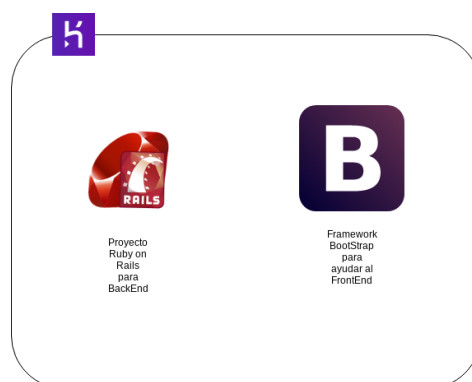


Figura 5.2: Desarrollo SE

Para llevar a cabo el proyecto se han empleado las siguientes herramientas:

- **Python:** como lenguaje de programación para la extracción de datos e implementación de algoritmos.
- **Clips:** como lenguaje para la creación de reglas del primer prototipo
- **Latex:** lenguaje para la documentación del proyecto. La plantilla utilizada ha sido desarrollada por el profesor Jesús Salido Tercero.
- **Ruby on Rails:** como lenguaje y framework para la aplicación web.
- **Heroku:** para despliegue de entornos en producción.
- **Numpy:** librería usada para el tratamiento de datos y creación de BBDD.
- **BeatifulSoup:** librería usada para la extracción de datos.
- **SciKit:** librería usada para el análisis de datos.
- **Vim:** Editor utilizado para RoR y LaTeX.
- **TexStudio:** IDE utilizado para compilar documentos en LaTeX.
- **Spyder:** IDE utilizado para compilación de Python.
- **Anaconda:** Framework utilizado para instalar los entornos de Python.
- **Github:** aplicación web utilizada para la creación y gestión de repositorios. Además también ha sido usada para tener un tablon kanvan.
- **Firefox:** navegador usado para la extracción de datos.
- **Draw.io:** aplicación web utilizada para la creación de gráficos, prototipos y diagramas.

5.1.7. ¿Qué nos quita el sueño por las noches?

El principal problema que nos podemos encontrar a la hora de realizar el proyecto es la situacionalidad de los casos a analizar. Puesto que aunque teóricamente sea correcto una acción frente a otra, dependerá de ambos tiradores que esta sea la correcta. Es por esto que se intentará hacer un sistema lo mas completo posible.

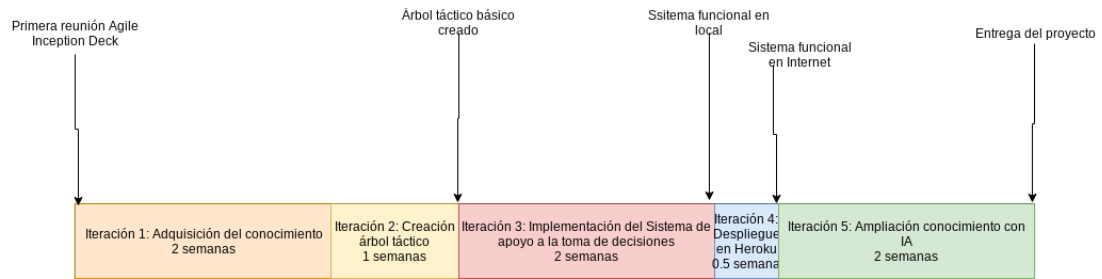
5.1.8. Tamaño del proyecto

Puesto que el sistema ha de ser bastante completo para tener una alta fiabilidad será complicado dar una aproximación precisa. Es por esto que se hará una estimación, teniendo muy presente que son estimaciones y puede variar. La estimación se dará en tiempo dedicado sin indicar fechas dada la disponibilidad de los recursos. La linea temporal del proyecto se puede ver en la figura 5.3 expuesta a continuación.

5.1.9. Muestra con claridad lo que se va a dar

Los objetivos a cumplir son los siguientes:

- Establecimiento del dominio del proyecto.
- Adquisición del conocimiento.

**Figura 5.3: Desarrollo SE****Tabla 5.1: Horas del proyecto**

| Etapas | Horas dedicadas por autor | Horas dedicadas por Experto |
|-------------|---------------------------|-----------------------------|
| Iteración 0 | 4h | 4h |
| Iteración 1 | 76h | 20h |
| Iteración 2 | 40h | 5h |
| Iteración 3 | 80h | 10h |
| Iteración 4 | 20h | 0h |
| Iteración 5 | 80h | 5h |
| Total | 300 | 44 |

- Representación del conocimiento con cliente Ruby on Rails para un fácil uso e interacción con el prototipo.
- Despliegue de la infraestructura del servicio en Heroku.

5.1.10. Muestra lo que va a conllevar

El análisis de costes del proyecto se llevará a cabo basándose a la planificación que se puede ver en el punto 8 (ver Figura 5.3). Los costes del proyecto se pueden dividir principalmente en dos tipos: personal e infraestructura.

El primer tipo, personal, dependerá del rol que se tome, puesto que dependiendo de este el coste a la hora será mayor. El primer rol que tendremos será el de desarrollador, el cual está estimado en 16€/h. Por otro lado tendremos el rol de experto, cuyo valor está estimado en 30€/h. En la tabla 5.1 se puede ver el coste de cada iteración, por rol y por horas. Además se podrá ver el coste total.

Por otro lado los costes de infraestructura han sido reducidos a cero. Puesto que no se busca lucrarse con el uso de herramienta, si no que sea útil para el resto se han buscado alternativas para reducir el coste disminuyendo en rendimiento de estas. Se ha optado por Heroku puesto que ofrece un plan totalmente gratuito a pesar de sus bajos recursos. Estos serán ampliables según las necesidades que se tengan.

Por lo que el coste total del proyecto puede verse en la tabla 5.2

Tabla 5.2: Costes del proyecto

| Concepto | Desglose | Horas | Coste |
|-----------------|--------------------|-------|-------|
| Personal | Autor del proyecto | 300h | 4800€ |
| | Experto | 44h | 1320€ |
| Infraestructura | Heroku project | - | 0€ |
| Total | 6120€ | | |

5.2. ITERACIÓN 1

Una vez establecido el tema a tratar en el proyecto y el alcance del mismo, se procede a desarrollar cada una de las iteraciones que lo componen.

En la primera iteración se ilustra el proceso de adquisición del conocimiento, lo que dará una visión mas detallada del proyecto y establecerá las bases para el desarrollo del árbol táctico que nos ayudará en la toma de decisiones en un asalto de esgrima.

5.2.1. Adquisición conocimiento básico

Lo primero que se hizo fue una investigación previa para asentar unas bases de conocimiento básicas sobre la materia en cuestión. Para ello se hicieron diversas búsquedas por internet buscando información acerca del deporte y su competición. Sacando bastante información en las páginas de la federación internacional y española. Además de diversos medios como el podcast *Llamada a pista* y redes sociales como *weareelgato* Explicar métodos de investigación y referencias para poder obtener unas nociones básicas sobre el dominio del tema.

Sobre estas búsquedas se pudo sacar información acerca del reglamento, el cual es algo básico de entender para saber como se puede plantear una estrategia. También se obtuvo información sobre los movimientos básicos de la esgrima, de este modo podríamos entender la jerga del deporte cuando nos reunieramos con los expertos. En cuanto al contenido multimedia se pudo sacar en claro como era un asalto de esgrima, afianzando todo el conocimiento obtenido anteriormente, viendo los movimientos que realizaban los tiradores y como se aplicaban las normas en los distintos casos. Además, se pudo ver como estos adaptaban sus estilos según transcurría el combate, por lo que nos hacía ver como se planteaba una táctica u otra en función de las características del asalto.

5.2.2. Adquisición conocimiento mediante entrevistas

Una vez que ya teníamos un conocimiento básico sobre la esgrima en la modalidad de espada, estábamos preparados para la primera entrevista con el experto. En la primera entrevista el objetivo principal no era más que poner en claro nuestros conceptos básicos para asegurarnos que podíamos entendernos con el experto, aunque no supieramos el razonamiento de cada caso, pero sí el que nos decía. Es por esto por lo que se decidió una entrevista abierta semi-estructurada.

La primera entrevista se dividiría en dos partes principales. La primera de ellas sería asegurar los conceptos adquiridos en la materia y aclarar los que no tuvieramos seguros. Además se corregirían aquellos que fueran erróneos. La segunda parte de la entrevista sería prácticamente abierta en su totalidad, ya que sería el experto el que nos tendría que introducir en la materia de escoger una táctica u otra. Para ello fuimos con unas preguntas básicas, las cuales son comunes a la mayoría de procesos de toma de decisiones.

Una vez finalizada la entrevista se analizarían los resultados y sacarían conclusiones de la misma. El documento de la primera entrevista se puede ver en el Anexo B.

Con la primera entrevista finalizada y después de analizar los resultados sacamos en claro las siguientes conclusiones:

- Lo primero en lo que hay que fijarse para planificar una estrategia es en la distancia, físico y experiencia.
- La distancia se valorará según la diferencia de altura entre tiradores y el puño usado de cada uno.
- La experiencia puede jugar a tu favor si eres el que más tiene.

- El físico influye a la hora de escoger como quieres llevar el asalto.
- La personalidad del tirador influye en su forma de tirar. Intentar averiguar como es esa persona antes de un asalto.
- No hay una preferencia ante ceder terreno o llevar al rival a final de pista. Dependerá de la situación del asalto.

Además sacamos las siguientes palabras para el glosario:

1. Cualidades del tirador
2. Distancia
3. Experiencia
4. Físico
5. Técnicas
6. Tocar
7. Entrar con hierro
8. Entrar con segunda intención
9. Altura
10. Puño
11. Francés
12. Anatómico
13. Más joven
14. Más rápido
15. Sentirse intimidado
16. Saber mucho
17. Ganar rápido
18. Echado hacia delante
19. Esperar
20. Provocar errores
21. Contra
22. Ser ofensivo
23. Alcance
24. Poco contacto en hierro
25. Perder distancia
26. Puntería

Tabla 5.3: Tabla objeto atributo y valor

| Objeto | Atributo | Valor |
|---------|------------|----------------------|
| Tirador | Puño | {Francés, Anatómico} |
| | Altura | [0, 230] cm |
| | Intimidado | {Si, No} |
| | Edad | [0, 120] Años |

- 27. Ser pasivo
- 28. Parar bien
- 29. Buen contrataque
- 30. Dejar corto
- 31. Dejar pensar

También actualizamos las tablas de objeto, atributo y valor (ver tabla 5.3)

Una vez reposado el conocimiento adquirido podremos pasar a la siguiente entrevista, de este modo ampliaremos la batería de posibilidades.

El planteamiento de la segunda entrevista era afianzar los conocimientos adquiridos en la primera y ampliar estos. Para ello lo primero que se hizo fue hacer preguntas abiertas sobre ciertas acciones en concreto. De esta manera nos aseguraríamos que estamos entiendo bien el contexto del problema. Después pasaríamos a ampliar el conocimiento, de nuevo con preguntas abiertas para conseguir desarrollar un esquema táctico mas amplio y complejo.

- El físico es importante pero no lo único a tener en cuenta. Los reflejos será algo muy a tener en cuenta junto a la velocidad de reacción.
- La teoría no lo es todo, también habrá que detectar como de bueno es el rival en las dos acciones principales de todo arte marcial, la defensa y el ataque. Con esto podremos trazar una táctica mas fiable y eficaz.
- La confianza en cada uno de los tiradores será determinante para saber que podemos movimientos podremos llevar a cabo o no.

Además sacamos las siguientes palabras para el glosario:

- 1. Coupé
- 2. Guardia
- 3. Defensa
- 4. Contra-ataque
- 5. Finta
- 6. Provocar
- 7. Engañar
- 8. Distraer
- 9. Llamada

Tabla 5.4: Tabla objeto atributo y valor

| Objeto | Atributo | Valor |
|---------|---------------------|---------------------|
| Tirador | Confianza | {Si, No} |
| | Reflejos | {Bajo, Medio, Alto} |
| | Velocidad | {Bajo, Medio, Alto} |
| | Capacidad defensiva | {Bajo, Medio, Alto} |
| | Capacidad ofensiva | {Bajo, Medio, Alto} |

10. Finta-Pase
11. Reflejos
12. Capacidad defensiva
13. Capacidad ofensiva
14. Confianza
15. Explosividad

También ampliamos las tablas de objeto atributo-valor. Ver tabla 5.4

El glosario final se puede ver en el anexo D. La tabla de objeto, atributo y valor final se puede ver en el anexo E.

5.3. ITERACIÓN 2

Una vez que ya tenemos conocimiento extraído de las entrevistas hay que plasmarlo. Para ello se obtendrán reglas mediante las cuales podamos tomar una decisión, es decir, crearemos un árbol de decisión el cual sea capaz de llevarnos a una táctica.

Para ello crearemos reglas simples, mediante las cuales con una serie de entradas, obtendremos una salida. Esta salida puede tener diversas formas. Puede ser una decisión final o la decisión de un camino intermedio para discernir si elegir una vía u otra. Estas reglas deberán ser supervisadas por el experto, ya que estas simularan su decisión.

La creación de estas reglas llevarán a un prototipo de sistema experto, el cual será la primera versión que podrá ser probada en un entorno local para ver si su uso aumenta el porcentaje de victorias, además del número de tocados dados y a su vez, disminuir el número de tocados recibidos.

Para ello se empezará con la creación de reglas básicas, como las de la primera entrevista. Esto nos dará un árbol de decisión que podrá ser ampliable en todo momento.

El primer paso para crear dicho árbol será tener claro cual es el proceso en la toma de decisión. Para ello se ha creado el diagrama, el cual ha sido corroborado por el experto. Dicho diagrama se puede observar en la figura 5.4.

Con el árbol de decisiones resumido, se elaboraron los sub-arboles para cada uno de los casos. Estos árboles han sido implantados como diagramas. En la figura 5.5 se puede ver el diagrama desarrollado para obtener la experiencia. El resto se puede ver en el anexo C donde se encuentran todos juntos.

Con los los flujos definidos en las tomas de decisión ilustrados mediante diagramas para favorecer su interpretación y corrección se puede empezar a desarrollar el sistema experto. Para ello inicialmente se hará una versión básica de éste mismo que se ampliará poco a poco. Se hará módulo por módulo para favorecer el desarrollo ágil y una vez listo cada módulo se implementará con el resto.

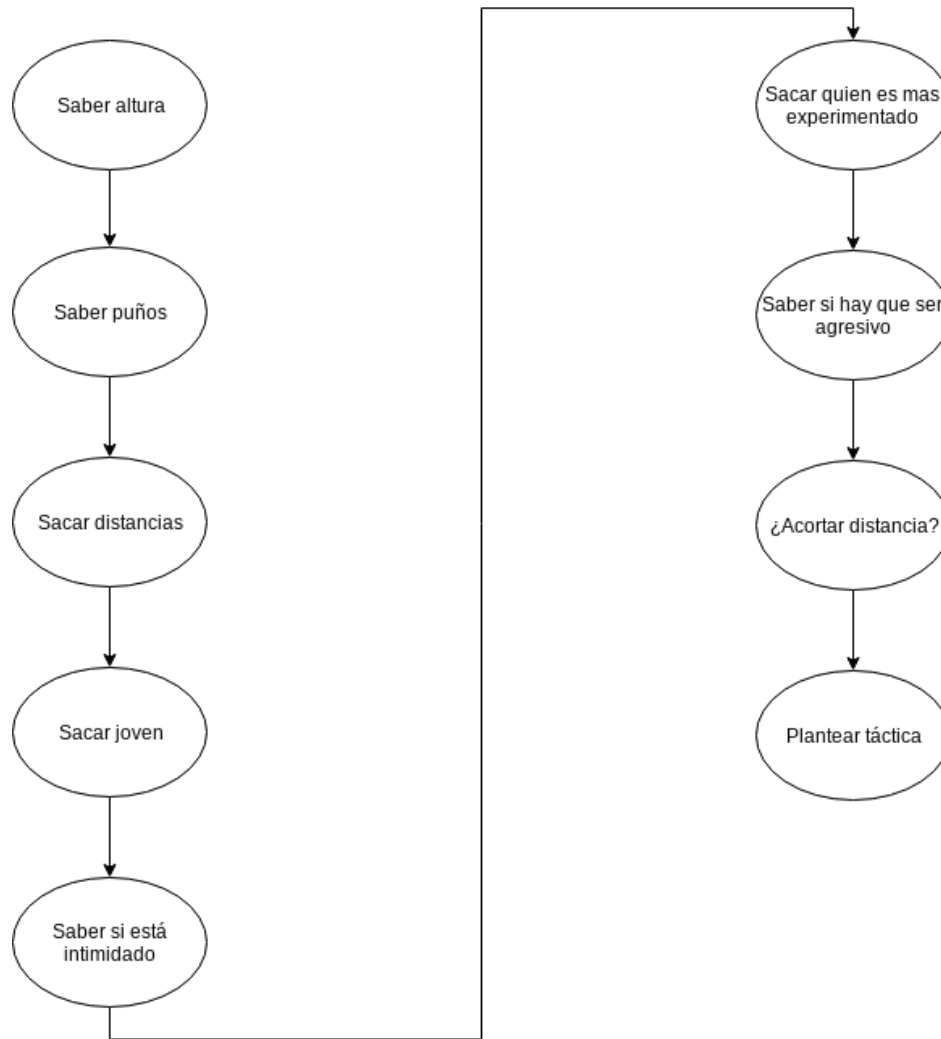


Figura 5.4: Esquema toma decisión

Al final obtendremos un prototipo del sistema experto desarrollado en CLIPS el cuál será funcional y podrá ser probado. De este modo conseguiremos pulir fallos en el proceso de desarrollo.

Para ello se han de definir reglas. En el proceso de definición de las reglas se usaron los diagramas desarrollados anteriormente. A continuación se muestra el desarrollo seguido para el módulo de experiencia. El diagrama de este módulo se puede ver en la figura 5.5 de este mismo capítulo.

Primero se estudió que variables harían falta. Para ello se definen tres variables las cuales serán booleanas indicando la altura. Solo una de ellas podrá ser verdadera y siempre habrá una verdadera. Dichas variables son *ALTOMASEL*, *ALTOIGUAL* y *ALTOMASYO*. Indicando la primera que el contricante es mas alto, la segunda que el primer tirador es mas alto mientras que la segunda que ambos son de la misma altura.

Para saber la empuñadura que usa cada tirador se han definido dos variables por tirador. De tal modo que quede como *EMPUÑADURAX* siendo empuñadura el tipo de empuñadura y X el tirador. de modo que las cuatro variables resultantes son: *FRANCESEL*, *FRANCESYO*, *ANATOMICOEL* y *ANATOMICOYO*.

Por último para definir quien tiene mas distancia, al igual que con la altura se han definido tres variables. Estas variables son *MASDISTANCIAEL*, *DISTANCIAIGUAL* y *MASDISTANCIAYO*. La primera indicaría que el rival tiene mas distancia, la segunda que ambos tienen la misma distancia, mientras que la última indicaría que el primer tirador tiene mas distancia.

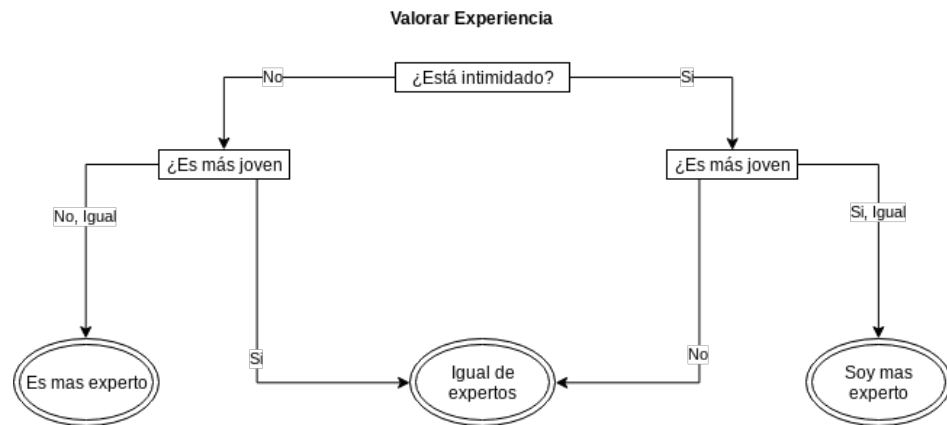


Figura 5.5: Árbol decisión experiencia

Con las variables definidas para cada nodo del árbol se definen las reglas, de tal modo que cuando estén todas definidas, todos los caminos del árbol estén cubiertos. En la tabla 5.5 se pueden ver las reglas junto al camino que cubren. Los caminos se pueden ver en la figura 5.6.

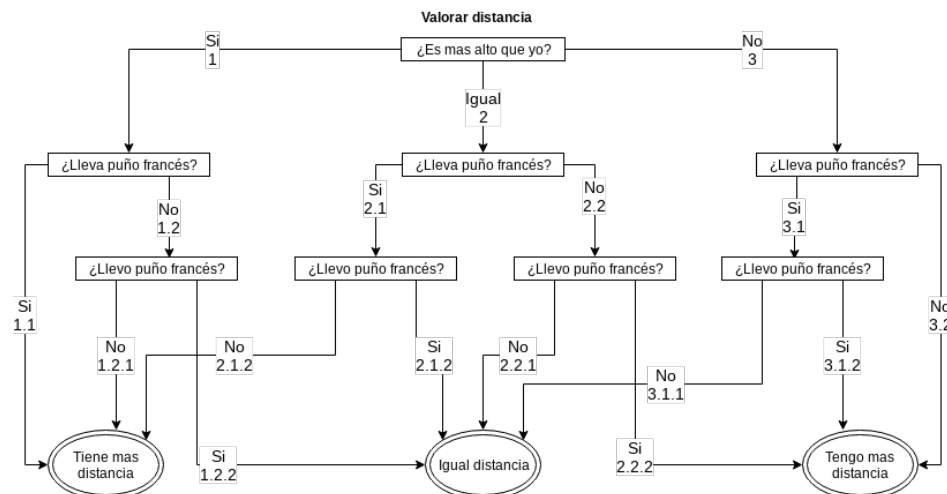


Figura 5.6: Arbol decisión distancia con caminos numerados

Con las reglas definidas las pasamos al sistema experto definidas en CLIPS.

A continuación se lista la plantilla y reglas utilizadas para el sistema experto en CLIPS.

```

1 (defrule cogerDatosTiradorEl "coger datos"
2
3   ?el <- (tirador (id el) (edad ?edadEl)(altura ?alturaEl)(puño ←
4     ← ?puñoEl)(intimidado ?intimidadoEl))
5   ?yo <- (tirador (id yo) (edad ?edadYo)(altura ?alturaYo)(puño ←
6     ← ?puñoYo))
7   ;(test (> ?edadEl ?edadYo))
8   =>
9   (assert
10     (alturaEl ?alturaEl)(alturaYo ?alturaYo)
11     (edadEl ?edadEl)(edadYo ?edadYo)
12     (puñoEl ?puñoEl)(puñoYo ?puñoYo)
13     (intimidadoEl ?intimidadoEl)
14   )
15   (retract ?el)
16   (retract ?yo)

```

Tabla 5.5: Reglas distancia

| Regla | Camino cubierto |
|--|-----------------|
| Si ALTOMASEL y FRANCESEL y FRANCESYO entonces MASDISTANCIAEL | 1.1 |
| Si ALTOMASEL y ANATOMICOEL y ANATOMICOYO entonces MASDISTANCIAEL | 1.2.1 |
| Si ALTOMASEL y FRANCESEL y ANATOMICOYO entonces MASDISTANCIAEL | 1.1 |
| Si ALTOMASEL y ANATOMICOEL y FRANCESYO entonces DISTANCIAIGUAL | 1.2.2 |
| Si ALTOIGUAL y FRANCESEL y FRANCESYO entonces DISTANCIAIGUAL | 2.1.2 |
| Si ALTOIGUAL y ANATOMICOEL y ANATOMICOYO entonces DISTANCIAIGUAL | 2.2.1 |
| Si ALTOIGUAL y FRANCESEL y ANATOMICOYO DISTANCIAMASEL | 2.1.2 |
| Si ALTOIGUAL y ANATOMICOEL y FRANCESYO entonces DISTANCIAMASYO | 2.2.2 |
| Si ALTOMASYO y FRANCESEL y FRANCESYO entonces DISTANCIAMASYO | 3.1.2 |
| Si ALTOMASYO y ANATOMICOEL y ANATOMICOYO entonces DISTANCIAMASYO | 3.2 |
| Si ALTOMASYO y FRANCESEL y ANATOMICOYO entonces DISTANCIAIGUAL | 3.1.1 |
| Si ALTOMASYO y ANATOMICOEL y FRANCESYO entonces DISTANCIAMASYO | 3.2 |

```

16 )
17
18
19 ;-----
20 ;
21 ;         Sacamos la altura
22 ;
23 ;-----
24
25
26
27
28 (defrule compararAltura1
29   (alturaEl ?x)(alturaYo ?y)
30   (test (> ?x ?y))
31   (test (> (- ?x ?y) 5)) ;seria para comprobar que la ←
      ↪ diferencia es mayor que 5
32   =>
33   (assert (altura masEl))
34 )
35
36 (defrule compararAltura1_2
37   (alturaEl ?x)(alturaYo ?y)
38   (test (> ?x ?y))
39   (test (<= (- ?x ?y) 5)) ;seria para comprobar que la ←
      ↪ diferencia es mayor que 5
40   =>
41   (assert (altura igual))
42 )
43
44 (defrule compararAltura2_1
45   (alturaEl ?x)(alturaYo ?y)
46   (test (< ?x ?y))
47   =>
48   (assert (altura masYo))
49 )
50
51 (defrule compararAltura2_2
52   (alturaEl ?x)(alturaYo ?y)
53   (test (<= ?x ?y))
54   (test (<= (- ?y ?x) 5)) ;seria para comprobar que la ←
      ↪ diferencia es mayor que 5
55
56   =>

```

Tabla 5.6: Sujetos prueba prototipo

| Número de sujeto | Edad | Altura | Años de experiencia | Empuñadura | Naturaleza |
|------------------|------|--------|---------------------|------------|------------|
| 1 | 38 | 160 | 1 año | Frances | Neutra |
| 2 | 43 | 168 | 1 año y 8 meses | Anatómico | Defensivo |
| 3 | 23 | 175 | 3 años y 6 meses | Anatómico | Agresivo |
| 4 | 35 | 183 | 10 años | Anatómico | Defensivo |

```

57 (assert (altura igual))
58 )

```

El resto de reglas se pueden ver en el anexo D.

Con el prototipo desarrollado era hora de ponerlo a prueba. Recordar que cada uno de los módulos fue revisado y aprobado por el experto. Para poner a pruebas el prototipo se llevó a la sala de esgrima de Ciudad Real. Para las pruebas se utilizaron sujetos cuyas características se pueden ver en la tabla 5.6.

5.4. ITERACIÓN 3

Con el prototipo del sistema experto desarrollado el siguiente paso fue proporcionar una interfaz amigable para el usuario a la par que accesible desde cualquier lugar.

Para ello lo primero fue crear unos bocetos sobre el diseño que se quería tener en la web a papel y boli. La idea original es que esta se componga de una página principal en la que se explique el objetivo de la página y cual es su proposito.

Por otro lado tendremos dos secciones que se diferencien la una de la otra, de las cuales una de ellas estará destinada a una versión rápida del sistema de apoyo mientras que la otra será una versión mas exhaustiva. De este modo nos podremos adaptar a las diversas situaciones en las que se puede utilizar la herramienta. La primera será para aquellas situaciones en las que tenemos menos de un minuto para que nos de un resultado. La segunda será para cuando podamos analizar los resultados e introducir las variables con mayor calma puesto que no tendremos prisa para ello.

Los resultados los bocetos se pueden ver en las figuras 5.7 y 5.8.

El resto de bocetos se pueden consultar en el anexo F.

Una vez con los bocetos claros se decidió que tecnologías usar. Para el desarrollo de la aplicación se decidió usar Ruby puesto que es uno de los lenguajes de programación mas utilizados para el desarrollo rápido de aplicaciones web. Otro de los motivos para escoger dicho lenguaje es el conocimiento obtenido por el desarrollador en dicho lenguaje, de este modo no tendremos que añadir tiempo de aprendizaje al proyecto, lo cual ahorrará recursos. En este caso el recurso será el tiempo que tarde en aprender el nuevo lenguaje.

El paradigma utilizado será Modelo-Vista-Controlador. Siendo este el que más favorece a la mantenebilidad del proyecto, permitiendo la ampliación de requisitos del mismo mediante modelos para después ampliar a usuarios, perfiles de tiradores, etc.

El desarrollo de la aplicación web se hizo en local, utilizando las herramientas que proporciona ruby para ello.

Para ello se siguieron los siguientes pasos:

1. **Inicialización:** Lo primero de todo será inicializar el proyecto. Hay muchos ejemplos de como inicializar un proyecto en ruby pero se siguió la documentación que te da la propia página web de ruby. Con esto tendremos lo básico para poder empezar a desarrollar la aplicación e ir añadiendo todo aquello que necesitemos.

Figura 5.7: Boceto página llenar

2. **Creación controladores:** Crearemos los primeros controladores de la aplicación que serán los mas importantes puesto que serán los encargados de manejar las peticiones hechas a las dos partes principales de la aplicación. Estos son los controladores para el apartado guiado y el completo. Además en este paso añadiremos los formularios en las vistas de dichos controladores de modo que podamos hacer las primeras pruebas.
3. **Creación servicio toma decisión:** Crearemos un servicio que será encargado de toda la lógica del sistema de apoyo a la toma de decisión. Aquí nos encargaremos de trasladar todo el conocimiento obtenido anteriormente a una clase que tenga toda la lógica de modo que este servicio sea accesible por toda la aplicación. Además en este apartado añadiremos los correspondientes test para comprobar que la implementación ha sido la adecuada.
4. **Creación vista resultados:** Daremos visualización a los resultados obtenidos por el servicio de toma de decisiones. De esta forma el usuario podrá ver cual es el resultado de dicho sistema.
5. **Añadir bootstrap:** Añadiremos el framework bootstrap para ayudarnos a la aplicación de estilos en la página. De esta manera mejoraremos la usabilidad de la misma.
6. **Añadir traducciones:** Haremos los cambios necesarios para permitir a la aplicación traducciones en distintos idiomas de forma que se pueda internacionalizar. En esta primera iteración se añadirá castellano e inglés.
7. **Añadir página de inicio:** Crearemos la página de inicio en la que daremos información general de la aplicación, cual es su objetivo y de donde surge.
8. **Añadir página de contacto:** Crearemos una página de contacto en la que se dará información de los responsables del proyecto y donde encontrarles.
9. **Añadir página de sugerencias:** Crearemos una página en la que los usuarios podrán plasmar sus sugerencias de modo que la página y el sistema tenga retro-alimentación y así perfeccionarlo.
10. **Añadir página entrenamiento:** Se añadirá un apartado en la página cuyo destino será que los tiradores puedan consultarla y obtener ideas de entrenamiento. Al principio tendrá una

MFT Inicio Llenar **Guiado** Contacto Sugerencias Entrenamiento Inglés

Formulario guiado sistema

¿Tienes mas distancia?

☐ Si

☐ No

☐ Igual

¿Eres mas joven?

☐ Si

☐ No

☐ Igual

Marque la siguiente opción si está intimidado

☐ Intimidado

Calcular

Figura 5.8: Boceto página guiado

serie de acciones con un seleccionador de estas aleatorio y así que puedan practicarlas forzando a realizar dichas acciones.

11. **Definir y añadir estilos:** Se definirán los estilos de los formularios, botones, textos, etc de la aplicación.

Con todo esto ya terminado podremos pasar al siguiente paso que será dar acceso a la aplicación a cualquier persona que tenga acceso a internet.

5.5. ITERACIÓN 4

Para darle una mayor accesibilidad al sistema se decidió desarrollar una aplicación web, de este modo cualquier persona con conexión a internet podría acceder a ella. El siguiente paso en el desarrollo del proyecto sería darle acceso a cualquier persona con internet.

Para ello tendrá que estar alojada en algún servidor externo. Puesto que no guardamos ningún dato sensible, no hace falta que esté alojado en Europa. Después de analizar las distintas posibilidades se decide usar Heroku. El análisis para esta decisión está en el anexo X (realizar anexo).

5.6. ITERACIÓN 5

Para ampliar el conocimiento y dar un mayor abanico de opciones a tener en cuenta en nuestro sistema de apoyo a la toma de decisión se intenta pronosticar quien será el ganador de un asalto, de esa forma el sistema podrá tenerlo en cuenta para dar una recomendación u otra.

Para el pronóstico se usarán técnicas de aprendizaje automático que se verán mas adelante. Para poder aplicarlas hace falta una BBDD. Después de una larga búsqueda fallida se decide crear nuestra propia BBDD sacando la información de la página de la federación internacional de esgrima (FIE).

5.6.1. Obtención BBDD

En dicha página disponemos de toda la información de cada torneo acontecido desde hace varios años con los resultados obtenidos en cada encuentro. Gracias a esto podemos recopilar toda la información

Tabla 5.7: Estructura BBDD Inicial

| Campo | Tipo | Descripción | Ejemplo |
|-------------------|---------|---------------------------------|-------------------------------|
| CompetitionID | String | Identificador de la competición | 2019-64 |
| Tableu | Integer | 32 | 32 |
| Competitor1 | String | Identificador del competidor 1 | /fencers/Anna-KOROLEVA-40351/ |
| Competitor2 | String | Identificador del competidor 2 | /fencers/Kira-KESZEI-49034/ |
| ResultCompetitor1 | String | Resultado del competidor 1 | V/15 |
| ResultCompetitor2 | String | Resultado del competidor 2 | D/13 |

Tabla 5.8: Ejemplo BBDD inicial

| CompetitionID | Tableu | Competitor1 | Competitor2 | ResultCompetitor1 |
|---------------|--------|-------------------------------|------------------------------|-------------------|
| 2019-64 | 32 | /fencers/Anna-KOROLEVA-40351/ | /fencers/Kira-KESZEI-49034/ | V/15 |
| 2019-64 | 32 | /fencers/Greta-CECERE-45345/ | /fencers/Andreea-LUPU-37410/ | V/15 |

necesaria para poder obtener conocimiento y transmitirlo a nuestro sistema para que este pueda nutrirse de él y que de este modo sea mas amplio y fiable.

Para la obtención de datos se han realizado diferntes scripts desarrollados en python mediante la técnica de scrapping, por lo que es posible que no sean funcionales cuando esté leyendo esto dado que la página podría haber cambiado el diseño y/o estructura, pero tan solo habría que adaptar parte de ellos para que vuelvan a ser funcionales. La técnica de scrapping se basa en visitar la página web y explorar su código para obtener la información de la misma.

Primero observamos la página y vimos como estaba estructurada. En este caso listaban los torneos dando información sobre ellos como el tipo de torneo, género, arma, categoría, etc. También usaban una paginación por lo que esto nos hacía pensar que había muchos tipos de torneos y desde hace bastantes años. Cuando nos metimos a ver la estructura de dichos torneos vimos como los registros eran diferentes, indicando solo la clasificación general algunos de ellos. Esto sucedía sobre todo cuanto mas tiempo tuvieran, por lo que se decidió hacer una primera criba y obtener aquellos a partir del 2015, que son aquellos en los que se empezaron a registrar los resultados de las eliminatorias además de la clasificación general del torneo. Esta es la información que nos interesa sacar, dado que con la clasificación general será algo más complicado sacar conocimiento de esta para nuestro objetivo final. Mientras que con los resultados obtenidos de cada enfrentamiento, junto a las características de cada tirador como pueden ser ranking FIE, mano usada, edad y nacionalidad podremos sacar algo de conocimiento con todo ello.

Con la primera criba hecha pasamos a sacar la información de cada torneo. Para ello exploraremos la página con la información de cada competencia la cual contiene la clasificación general, la cual no nos interesa almacenar. Por otro lado tenemos la fase de grupos o también conocida como poules. Esta fase también la ignoraremos de momento pudiendo ser esta información a analizar en un futuro. Ya sólo nos quedarían las fases eliminatorias o cuadros de directas. Estos contienen el resultado del enfrentamiento: tocados dados por cada tirador, quien ganó, el identificador de cada tirador y el tablón que se disputaba. Con esto la estructura de la BBDD se queda de la siguiente forma (ver tabla 5.7):

A continuación en la tabla 5.8 se puede ver un ejemplo del estado inicial de la BBDD.

Con esta información recopilada nos faltaría completarla. Para ello extraimos en una BBDD aparte todos los identificadores de los competidores, de esa forma podríamos extraer sus características en una tabla aparte la cual recopile la información de todos los competidores. Esta tabla contendrá la siguiente información de cada tirador: identificador, edad, ranking, nacionalidad, mano dominante y arma. De tal forma que la estructura se puede ver en la tabla 5.9 y un ejemplo en la tabla 5.10.

Con toda la información recopilada juntarmeos ambas BBDD de forma que cruzando los datos de ambas podremos sustituir los identificadores del de los tiradores de la primera BBDD con los datos

Tabla 5.9: Estructura BBDD tiradores

| Campo | Tipo | Descripción | Ejemplo |
|--------------|---------|--|----------------------|
| ID | String | Identificador del tirador | ADRIANA-MILANO-36467 |
| Edad | Integer | Edad del tirador | 32 |
| FieRanking | String | Ranking de campeonatos FIE del tirador | 1234 |
| Nacionalidad | String | Nacionalidad del tirador | VENEZUELA |
| Mano | String | Mano dominante del tirador | Right |
| Arma | String | Arma principal del tirador | Sabre |

Tabla 5.10: Ejemplo BBDD tiradores

| ID | Edad | FieRanking | Nacionalidad | Mano | Arma |
|-------------------------|------|------------|--------------|-------|-------|
| ADRIANA-MILANO-36467 | 21 | None | VENEZUELA | Right | Sabre |
| AKHMETOV-Iskander-35108 | 22 | 82 | RUSSIA | Right | Foil |

recopilados de cada tirador.

De esta forma conseguiremos tener toda la información en una sola BBDD para que los datos puedan ser comparados con mayor facilidad. La estructura final de la BBDD se puede ver en la tabla 5.11

5.6.2. Tratamiento BBDD

El primer paso que daremos será añadir una columna numérica indicando quien fue el ganador del encuentro. Esta contendrá 0 o 1 dependiendo de si ganó el primer tirador o el segundo respectivamente. De esta forma será mas fácil identificar quien de los dos tiradores ganó. La estructura del nuevo campo se podrá ver en la tabla 5.12. Para la obtención de este campo se exploró el asalto correspondiente y se comprobó quien tenía dentro de su resultado la letra "V" dado que esta es la que indicaba la victoria.

Una vez calculado este campo ya no nos harían falta los campos de resultados por lo que se procedió al borrado de los mismos.

El siguiente cambio que haremos será normalizar la BBDD. Esto se hace para que los datos puedan ser comparados de una forma mas eficiente. Para ello transformamos aquellas variables de texto a numéricas, de este modo serán mas sencillos los cálculos estadísticos.

En nuestro caso tenemos la mano usada por el tirador y el arma usada por este. Para ello seguiremos la transformación mostrada en la tabla 5.13 y 5.14 para las equivalencias de mano y arma respectivamente. Esto se resume en cambiar la mano dominante diestra por un valor de 0 mientras que la zurda será sustituida por el valor 1. También cambiaremos las armas siendo el equivalente de espada el 2, 1 para florete y restando el 0 para sable.

El siguiente paso era eliminar aquellas columnas que no nos sirvan. Para ello primero quitaremos aquellas que puedan introducir mucho ruido y que son fáciles de detectar como por ejemplo el identificador del tirador, puesto que es su nombre. Este no tendrá ninguna relación con el resultado. Del mismo modo quitaremos las columnas de resultados de resultados de ambos tiradores puesto que ya tenemos la de victoria. Otro campo que está en la misma situación será el identificador de la competición.

Después de esto la estructura de la BBDD se puede ver en la tabla 5.15.

5.6.3. Extraer conocimiento de los datos

Una vez que tenemos la BBDD normalizada el siguiente paso será buscar relaciones entre los datos de modo que se pueda extraer conocimiento de estos mismo y de esa forma poder mejorar nuestros

Tabla 5.11: Estructura BBDD final

| Campo | Tipo | Descripción | Ejemplo |
|----------------|---------|-------------------------------------|-----------------------------|
| ID | String | Identificador de la competición | ADRIANA-MILANO-36467 |
| TABLEU | Integer | Tablón en el que se juega el asalto | 32 |
| C1_ID | String | Identificador del primer tirador | Alexander-CHOUPENITCH-21765 |
| C1_AGE | Integer | Edad del primer tirador | 32 |
| C1_RANKING | Integer | Ranking del primer tirador | 10 |
| C1_NATIONALITY | String | Nacionalidad del primer tirador | CZECH REPUBLIC |
| C1_HANDNESS | String | Mano dominante del primer tirador | Right |
| C1_WEAPON | String | Arma principal del primer tirador | Epee |
| C2_ID | String | Identificador del segundo tirador | Alexander-CHOUPENITCH-21765 |
| C2_AGE | Integer | Edad del segundo tirador | 32 |
| C2_RANKING | Integer | Ranking del segundo tirador | 10 |
| C2_NATIONALITY | String | Nacionalidad del segundo tirador | CZECH REPUBLIC |
| C2_HANDNESS | String | Mano dominante del segundo tirador | Right |
| C2_WEAPON | String | Arma principal del segundo tirador | Epee |
| RESULT_C1 | String | Resultado del primer tirador | V/15 |
| RESULT_C2 | String | Resultado del segundo tirador | D/7 |

Tabla 5.12: Estructura campo ganador

| Campo | Tipo | Descripción | Ejemplo |
|--------|---------|---|---------|
| WINNER | Integer | Este campo nos indicará quien ganó. Siendo 0 el primer tirador y 1 el segundo | 1 |

Tabla 5.13: Tabla equivalencias mano

| Valor antiguo | Valor nuevo |
|---------------|-------------|
| Right | 0 |
| Left | 1 |

Tabla 5.14: Tabla equivalencias mano

| Valor antiguo | Valor nuevo |
|---------------|-------------|
| Sabre | 0 |
| Foil | 1 |
| Epee | 2 |

Tabla 5.15: Estructura BBDD final

| Campo | Tipo | Descripción | Ejempl |
|----------------|---------|---|--------|
| TABLEU | Integer | Tablón en el que se juega el asalto | 32 |
| C1_AGE | Integer | Edad del primer tirador | 32 |
| C1_RANKING | Integer | Ranking del primer tirador | 10 |
| C1_NATIONALITY | String | Nacionalidad del primer tirador | CZECH |
| C1_HANDNESS | String | Mano dominante del primer tirador | Right |
| C1_WEAPON | String | Arma principal del primer tirador | Epee |
| C2_AGE | Integer | Edad del segundo tirador | 32 |
| C2_RANKING | Integer | Ranking del segundo tirador | 10 |
| C2_NATIONALITY | String | Nacionalidad del segundo tirador | CZECH |
| C2_HANDNESS | String | Mano dominante del segundo tirador | Right |
| C2_WEAPON | String | Arma principal del segundo tirador | Epee |
| WINNER | Integer | Este campo nos indicará quien ganó. Siendo 0 el primer tirador y 1 el segundo | 1 |

sistema. El objetivo inicial es el de obtener un pronóstico de quien ganará en un enfrentamiento directo. De este modo podremos completar nuestro sistema adaptando una táctica mas defensiva en función del resultado. No podremos ser agresivos o intentar movimientos que no tengamos afianzados ante un rival que es muy probable que nos gane. En este caso mencionado antes lo mejor será asegurar y llevarnos el asalto a nuestro terreno.

Como hemos mencionado antes el problema que tenemos antes es el de intentar pronosticar el vencedor ante un enfrentamiento entre dos tiradores. Eso denota a que los resultados serán ganador uno o ganador otro. Por ende nos enfrentamos ante un problema de clasificación binaria. Clasificación puesto que los resultados estarán en grupos y binario porque el número de posibles opciones son dos en este caso en concreto denotado por el objetivo final.

Además podremos aplicar técnicas de aprendizaje supervisado, dado que tenemos un conjunto de datos previamente etiquetados gracias a todo el trabajo realizado anteriormente. Además podemos estar seguros de que el etiquetado es el correcto, puesto que la veracidad de estos datos es completa. Ante un problema de tales características los algoritmos más utilizados suelen ser arboles de decisión, clasificación de Naïve Bayes, regresión por minimos cuadrados, regresión logística, Support Vector Machines (SVM) y métodos ensemble (conjuntos de clasificadores).

Para elegir alguno de estos podemos seguir la chuleta que nos proporciona scikit-learn la cuál se puede encontrar en la figura 5.7

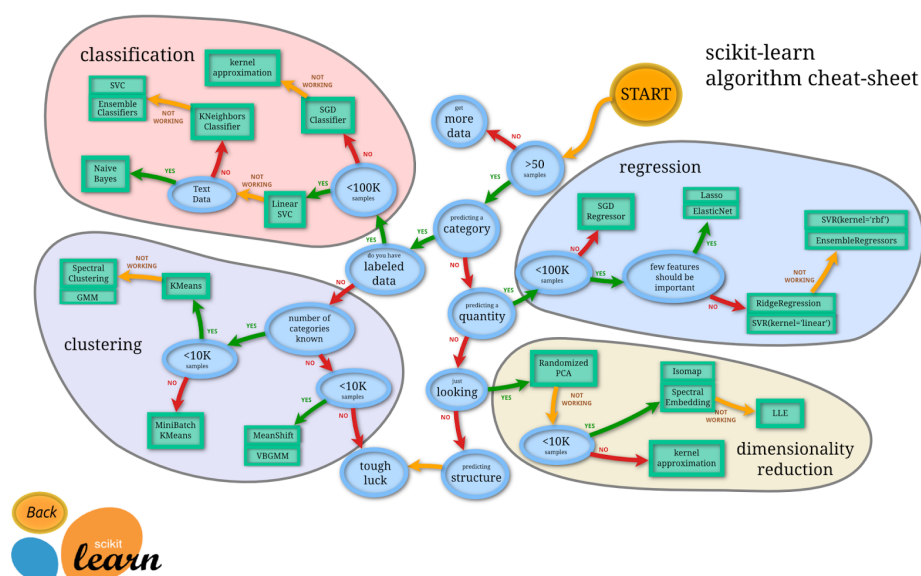


Figura 5.9: Chuleta decisión algoritmo ML

Siguiendo esta chuleta y siguiendo el flujo veremos que algoritmo nos recomienda. Para la primera pregunta la respuesta sería sí, dado que tenemos mas de 50 ejemplos. Como hemos visto antes intentamos predecir una categoría y además tenemos los datos etiquetados, las dos siguientes respuestas serán afirmativas. La siguiente pregunta es que si tenemos menos de 100.000 ejemplos y en este caso también será afirmativa. Nuestra BBDD se compone de 25.045 registros.

Para aplicar los algoritmos usaremos la librería scikit-learn dado que nos proporciona todas las herramientas que necesitaremos para aplicarlos.

El primer algoritmo que deberemos aplicar según la chuleta será linear SVC. En caso de que no funcione seguiremos con la chuleta.

Después de varias pruebas el mejor resultado obtenido fue de un 63.78 % de acierto. Es algo bajo por lo que nos disponemos a buscar alternativas. El código utilizado fue el siguiente:

```
1 LSVC = LinearSVC(dual=False, fit_intercept=False)
```

```

2
3 LSVC.fit(x1,y1)
4 y2_LR_model = LSVC.predict(x2)
5 print("LR Accuracy :", accuracy_score(y2, y2_LR_model))

```

El siguiente a probar será un clasificador por cercanía de vecinos o mas conocido como KNN.

Para ello lo primero será averiguar el número de vecinos óptimo. El plan que seguimos fue probar diferentes opciones y ver cual era la que nos daba una mayor precisión. El resultado que obtuvimos fue que el mejor valor era de 53 con una precisión del 0.6460194963444355 %. En la figura X.x se puede ver la gráfica comparativa.

Este fue el código utilizado:

```

1 iterations = range(0, 10)
2 for i in iterations:
3     k_range = range((i * 25) + 1, (i * 25) + 26)
4     scores = []
5     for k in k_range:
6         knn = KNeighborsClassifier(n_neighbors=k, n_jobs=5)
7         knn.fit(X_train, y_train)
8         y_pred = knn.predict(X_test)
9         scores.append(metrics.accuracy_score(y_test, y_pred))
10
11     if max_score < max(scores):
12         max_score = max(scores)
13         max_k = scores.index(max(scores)) + (i * 25)
14
15     plt.plot(k_range, scores)
16
17 plt.xlabel('Value of K for KNN')
18 plt.ylabel('Testing Accuracy')
19 plt.show()
20 plt.close()
21 print ("max_score:")
22 print (max_score)
23 print ("max_k:")
24 print (max_k)

```

El siguiente parámetro a probar será el número de hijos que deberá tener cada hoja en la exploración. Para ello lanzaremos el siguiente código cuyo resultado se puede observar en la figura X.x. El resultado que obtuvimos fue que el mejor valor estaba entre 29 y 57, por lo que nos quedaremos con un valor de 30 puesto que es el de por defecto. Este valor nos proporcionaba una precisión de 0.6436839967506093 %. El código utilizado fue el siguiente:

```

1 k_range = range(1, 60)
2 scores = []
3
4 max_score = 0
5 max_k = 0
6
7 for i in k_range:
8     knn = KNeighborsClassifier(n_neighbors=53, n_jobs=5, ←
9         ↪ leaf_size=i)
10    knn.fit(X_train, y_train)
11    y_pred = knn.predict(X_test)
12    score = metrics.accuracy_score(y_test, y_pred)
13    scores.append(score)
14    if max_score < score:
15        max_score = score
16        max_k = i

```

```

17 plt.plot(k_range, scores)
18 plt.xlabel('Value of leaf_size for KNN')
19 plt.ylabel('Testing Accuracy')
20 plt.show()
21 plt.close()
22
23 print ("max leaf size")
24 print ("max_score:")
25 print (max_score)
26 print ("max_k:")
27 print (max_k)

```

La siguiente variable a probar sería el valor de P, el cuál representará el power. Repetiremos el mismo proceso para esta variable. El mejor resultado lo obtuvimos con el valor de 1 con un porcentaje de 0.6487611697806661 %. La gráfica se puede observar en la figura X.x y el código utilizado es el siguiente:

```

1 k_range = range(1, 100)
2 scores = []
3
4 max_score = 0
5 max_k = 0
6
7 for i in k_range:
8     knn = KNeighborsClassifier(n_neighbors=53, n_jobs=5, ←
9         ↪ leaf_size=30, p=i)
10    knn.fit(X_train, y_train)
11    y_pred = knn.predict(X_test)
12    score = metrics.accuracy_score(y_test, y_pred)
13    scores.append(score)
14    if max_score < score:
15        max_score = score
16        max_k = i
17
18 plt.plot(k_range, scores)
19 plt.xlabel('Value of p for KNN')
20 plt.ylabel('Testing Accuracy')
21 plt.show()
22 plt.close()
23
24 print ("max p value")
25 print ("max_score:")
26 print (max_score)
27 print ("max_k:")
28 print (max_k)

```

Finalmente el modelo tendrá una precisión de 0.6487611697806661 % el cuál es algo aceptable para los datos que tenemos. El modelo es el siguiente:

```

1 knn = KNeighborsClassifier(n_neighbors=53, n_jobs=5, ←
2     ↪ leaf_size=29, p=1)
3 knn.fit(X_train, y_train)
4 y_pred = knn.predict(X_test)
5 print (metrics.accuracy_score(y_test, y_pred))

```

Puesto que se podría llegar a encontrar una solución mas óptima pero habría que probar por fuerza bruta preparamos un algoritmo que pruebe estas tres variables entre los rangos que vimos que podrían ser válidos. Estos rangos son de 1 a 2500 para número de vecinos, de 1 a 60 para el tamaño de hoja y de 1 a 20 para el valor de p.

Dicho algoritmo lo podemos encontrar en el anexo G.

Tabla 5.16: Parámetros petición predicción

| Parámetro | Descripción | Tipo | Requerido | Valores admitidos | Ejemplo |
|------------------|---|---------|-----------|----------------------------|---------|
| tableu | Tablón encuentro | Integer | Si | {2, 4, 8, 16, 32, 64, ...} | 2 |
| fencer1_age | Edad tirador 1 | Integer | Si | {1, inf} | 26 |
| fencer1_ranking | Ranking tirador 1 | Integer | Si | {1, inf} | 33 |
| fencer1_handness | Mano tirador 1 (0 derecha, 1 izquierda) | Integer | Si | [0, 1] | 0 |
| fencer1_weapon | Arma tirador 1 (0 florete, 1 sable, 2 espada) | Integer | Si | [0, 1, 2] | 1 |
| fencer2_age | Edad tirador 2 | Integer | Si | {1, inf} | 44 |
| fencer2_ranking | Ranking tirador 2 | Integer | Si | {1, inf} | 9999999 |
| fencer2_handness | Mano tirador 2 (0 derecha, 1 izquierda) | Integer | Si | [0, 1] | 0 |
| fencer2_weapon | Arma tirador 2 (0 florete, 1 sable, 2 espada) | Integer | Si | [0, 1, 2] | 1 |

Los resultados obtenidos se consiguieron después de mas de dos días de procesamiento. Para el valor de K se obtuvo que el valor de 56, siendo este el número de grupos. Por otro lado el tamaño máximo de hoja será de 29. Por último el valor de p será de 5. Con esto se consiguió una precisión de 0.652315190901706 %, consiguiendo una mejora de 0.05 puntos de precisión.

Esto mereció la pena debido a que teníamos bastante tiempo para lanzar el algoritmo. Puesto que los resultados ya se han obtenido y son mejores que los obtenidos con anterioridad nos quedaremos con ellos.

5.6.4. Usar conocimiento en aplicación

Con el modelo optimizado lo máximo posible y obtenido un resultado mas que aceptable para los datos que manejamos podremos incluirlo en nuestro sistema. Para ello aprovecharemos que el modelo nos proporciona el porcentaje de probabilidad de que gane cada uno, lo cual nos dará la oportunidad de poder crear reglas en función de las probabilidades. Para poder acceder a ello crearemos una API a la cual llamaremos cada vez que queramos realizar la consulta.

Esta API estará realizada en python con flask la cual solo tendrá una llamada que devolverá la probabilidad de que gane cada uno de los tiradores. En los parámetros de esta API estarán incluidas las características del asalto, es decir, características de cada tirador y tablón. De esta forma podremos llamar al modelo, que este haga la predicción y que nos devuelva el resultado.

El endpoint llamado con curl será de la siguiente manera:

```
1 curl -X GET https://mftapi.herokuapp.com/predict/ -d ↵
    ↵ "tableu=16&fencer1_age=26&fencer1_ranking=33& ↵
    ↵ fencer1_handness=0&fencer1_weapon=1.0&fencer2_age=44& ↵
    ↵ fencer2_ranking=99999999&fencer2_handness=0& ↵
    ↵ fencer2_weapon=1.0"
```

Los parámetros con sus características se podrán consultar en la tabla 5.16

Para la implementación de dicha API se utilizó el lenguaje de programación Python 3.6.9 para aprovechar el código utilizado a la hora de entrenar el modelo anterior. Las librerías utilizadas para las peticiones fueron Flask junto a Unicorn para darle una interfaz web.

Por otro lado el hosting utilizado volvió a ser Heroku dado que tiene un límite de cinco proyectos por cuenta de forma gratuita. Esto nos permite ahorrar costes en el proyecto Además del endpoint mencionado anteriormente tendremos otro para que haga una llamada vacía la cual será la encargada de levantar el servidor, puesto que el programa gratuito apaga las máquinas a los treinta minutos de no recibir peticiones. Cuando accedan a la página, una vez por sesión hará la llamada asincrónica a dicho endpoint, de este modo la máquina de la API se irá levantando mientras el usuario navega por la web o rellena el formulario. De este modo bajaremos el tiempo de espera del usuario de treinta

Tabla 5.17

| Objeto | Atributo | Valor |
|---------|---------------------------------|------------|
| Tirador | Probabilidad victoria tirador 1 | [0, 100] % |

segundos aproximados que tarda la máquina en estar lista a esos treinta segundos menos el tiempo que tarde en rellenar el formulario, siendo este una media de 20 segundos. El mínimo de tiempo que estará el usuario esperando será de tres segundos puesto que es lo que tarda el endpoint de predicción en llevar a cabo la petición.

El código utilizado se puede ver en el anexo G.

Una vez con todo este proceso hecho, se mantuvo una conversación con el experto mediante mensajería instantánea. En dicha conversación se volvieron a destacar conceptos ya mencionados anteriormente como la confianza del tirador, tanto la de uno mismo como la del rival. También menciono cuanto había que arriesgar en función de la diferencia de nivel entre ambos tiradores. Pudiendo arriesgar más cuanto mas haya, siempre que esté a tu favor. En caso contrario se procurará ser lo mas conservador posible. Al ser conservadores el tirador irá cogiendo confianza con los tocados que mejor le salen y de esta forma podrá llevar el asalto a su terreno.

Esto nos llevó a ampliar las variables del sistema experto añadiendo un parámetro más: diferencia de nivel. Este parámetro será un número entre 0 y 100 representando porcentajes de probabilidad de que gane el primer tirador.

Esta variable será tenida en cuenta para el siguiente árbol que representará cuanto riesgo ha de tomar un esgrimista en la pista. Dicho árbol se puede ver en la figura 5.10

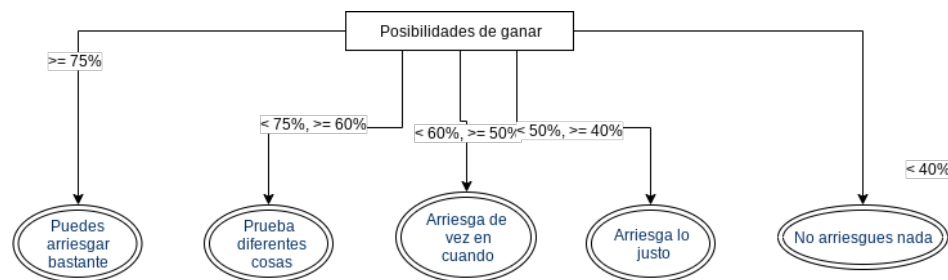


Figura 5.10: Mapa conocimiento arriesgar

La representación del mapa es que si la estadística juega totalmente a nuestro favor podremos arriesgar bastante, dado que en teoría podremos tomar las riendas del asalto en cualquier momento. Si se decantan hacia nuestro lado, pero no del todo nos podremos permitir arriesgar de vez en cuando, pero sin tomarnos demasiadas licencias, puesto que nos costará mas remontar el asalto si se complica. En caso de que las cosas estén empatadas podremos probar de vez en cuando, pero siendo la mayor parte del tiempo conservadores. En caso de que se decanten un poco hacia su lado las cosas tendremos que arriesgar en las ocasiones mas claras. Por último si se decantan hacia su lado no podremos arriesgar nada, puesto que lo normal es que cualquier error sea pagado muy caro.

CONCLUSIONES

En este capítulo se realizará un juicio crítico y discusión sobre los resultados obtenidos. Si es pertinente deberá incluir información sobre trabajos derivados como publicaciones o ponencias, así como trabajos futuros, solo si estos están planificados en el momento en que se redacta el texto. Además incluirá obligatoriamente la explicación de cómo el trabajo realizado satisface las competencias de la tecnología específica cursada.

6.1. OBJETIVOS ALCANZADOS

El objetivo con el que se inicio este proyecto fue desarrollar un sistema el cual fuera capaz de ayudar a aquellas personas que se iniciaran en mundo de la esgrima. Cuando empiezas es difícil comprender que está pasando y en tus primeras competencias sin nadie que te guíe en cada asalto hace que la curva de aprendizaje e introducción al deporte sea demasiada inclinada. Desde el inicio se descarto desarrollar el sistema completo puesto que este sería muy complejo y llevaría mas tiempo del disponible para realizar este proyecto, por eso se decidió desarrollar un prototipo del sistema el cual cubra el arma mas utilizada. A continuación se muestra el resumen de los objetivos planteados desde el inicio y cuanto han sido satisfechos:

- **Adquisición conocimiento:** El desarrollo de esta etapa no fue muy complicada puesto que se partía con una base teórica lo cuál facilitó el cumplimiento de este objetivo. También se contaba con una buena relación y disposición del experto, siendo este un factor fundamental para el éxito de este objetivo, facilitando este una gran disposición y apoyo en el proyecto desde el principio del proyecto. A destacar que no se captó todo el conocimiento necesario para desarrollar el arbol táctico completo de espada.
- **Creación arbol táctico:** La creación del arbol táctico fue todo un éxito gracias a la gran disposición por parte del experto, consolidando el conociendo adquirido anteriormente. Además aportó ideas que no fueron recogidas con anterioridad. Al final quedó muy claro y completo de acuerdo al marco con el que se trabajaba, siendo conscientes del tiempo disponible. En definitiva, se abordó todo el conocimiento adquirido durante el anterior objetivo y es por esto que se puede decir que fue todo un éxito.
- **Implantación sistema de apoyo a la toma de decisiones:** El desarrollo de este objetivo fue quizás el mas sencillo de todos por el perfil del desarrollador del proyecto. Desde el inicio se tenía claro que era un prototipo, por tanto no requería de grandes diseños y los bocetos fueron de gran importancia para tener una idea inicial de que se quería hacer. Esto facilitó el desarrollo de la implantación del sistema. Por otro lado la buena definición de requisitos también ayudo a tener una idea global de que se quería hacer, asimismo como una idea de que se quería hacer en cada paso. Por otro lado, el prototipo es bastante mejorable en algunos aspectos que se mencionan mas adelante y es por esto que no podemos decir que se haya cumplido por completo con este objetivo.

- **Accesible desde cualquier parte:** Este objetivo fue cumplido en su mayoría, puesto que se hizo un despliegue con Heroku para que sea accesible en cualquier sitio con internet, el cual será la mayoría de los casos, pero se han dejado algunos casos como cuando no dispongamos de una conexión a internet. También el uso desde otro dispositivo que no sea un ordenador de escritorio.
- **Ampliación con IA:** Este fue, con diferencia, el apartado mas costoso del proyecto. Desde la búsqueda de una BBDD que al final se tuvo que generar scrapeando los datos. Primero por rehacer el código varias veces debido a la remodelación de la página durante el desarrollo del mismo. Después la poca información disponible hizo que los datos no fueran suficientes para poder obtener conocimiento de estos, por lo que hubo que pasar de nuevo por el proceso de búsqueda y obtención de mas datos para que el conjunto de todos fueran relevantes. Durante este proceso hubo una nueva remodelación, por lo que prácticamente hubo que empezar de cero. Después hubo que exprimir al máximo para poder obtener unos resultados válidos y aún así no fueron perfectos. Se puede estar satisfecho con el resultado obtenido y la base que se partía, pero estos están lejos de ser perfectamente válidos. Es por todo esto que se puede decir que se ha cumplido parcialmente este objetivo.

Con todo lo mencionado anteriormente y viendo los resultados del proyecto se podría decir que ha sido un éxito, pero no por completo. Se han alcanzado todos los objetivos propuestos, pero el resultado de estos han sido mejorables.

6.2. TRABAJO FUTURO

Como se indicó al comienzo de este documento este sistema no es más que un prototipo. Es por esto que es muy susceptible a mejoras debido a que no está completado. En este apartado hablaremos de las debilidades que presenta actualmente el sistema. Después hablaremos sobre las principales mejoras que se podrían hacer a este para subsanar dichas flaquezas, además de otras para ampliar y mejorar el sistema. Este proyecto estará en continua evolución debido a que el conocimiento y las tácticas variarán con los años como se ha podido ver en la historia desarrollando diferentes estilos en la esgrima. También se crearán nuevos equipos como los puños, lo cual hará que haya que adaptarse a ellos. Asimismo las reglas van cambiando como este año cambió la pasividad, lo cual hace que haya que cambiar ciertas partes de la estrategia, permitiendo hacer cosas nuevas o evitando que se hagan otras.

6.2.1. Carencias

1. **Limitación armas:** La primera carencia que nos encontramos es la limitación en cuanto a las armas de este deporte. Solamente contamos con una de las tres existentes. A pesar de hacer el sistema para la mas popular, no se ofrece ayuda a todo el público posible, por lo que se deberá trabajar en ampliar para poder cubrir las tres armas.
2. **Tiempos carga:** A pesar de desarrollar dos modos para facilitar y agilizar los tiempos para cuando necesitemos una respuesta mas rápida (recordemos que los descansos entre asaltos son de un minuto) para la versión completa necesitamos, en el mejor de los casos, mínimo entre tres y cuatro segundos para que se calcule el resultado. Esto es debido a que la llamada la API tarda ese tiempo en calcular la predicción. Esto puede ser un problema para usuarios con poca paciencia. Otro de los problemas relacionados con los tiempos es el primer acceso a la aplicación. El plan gratuito de Heroku apaga los servidores al no recibir peticiones en treinta minutos, lo cual hace que se tengan que levantar de nuevo en el primer acceso, tardando estos unos 25 segundos.

3. **Experiencia usuario web:** La aplicación web ha sido desarrollada en poco tiempo debido al límite de horas dispuestas para este TFG. Otro factor a tener en cuenta es que el desarrollador de la aplicación no es experto en desarrollar interfaces. Por los motivos expuestos anteriormente el usuario final puede no tener la mejor experiencia y actualmente es bastante mejorable, desde los formularios haciendo que estos sean mas intuitivos, rápidos y explicativos por si mismos. No solo los formularios deberán ser repasados y mejorados, también la forma en la que se dan los resultados, los colores, la distribución del menú, etc.
4. **Limitación web:** Solo se ha desarrollado una aplicación web y no fue adaptada para móvil por falta de tiempo. Uno de los usos de la aplicación es en competición, por lo que no podrás estar con un ordenador portátil con conexión a internet. Es por esto que no se verá del todo bien desde el navegador del móvil y no se cubren los casos en los que no tengamos internet, como cuando viajemos al extranjero y no contemos con una tarifa para poder navegar por internet.
5. **Predicción:** El sistema de predicción terminó con un porcentaje aproximado de 65 % de acierto, es por esto que se deberá tener en cuenta debido a que es mejor que tirar una moneda al aire, pero su fiabilidad no esta cerca de ser totalmente fiable.
6. **Internacionalización:** La aplicación actualmente solo está disponible en dos idiomas, cierto es que de los mas comunes y utilizados no se llega a todos los tiradores de todos los paises. Faltan idiomas en los que la esgrima esta muy presente como húngaro, francés e italiano.

A continuación se hablará sobre las mejoras que se podrán hacer en cada apartado del sistema, dando una primera aproximación sobre la solución a tener en cuenta y, en caso de que corresponda, que problema subsana.

6.2.2. Sistema experto

- **Lógica difusa:** Añadiendo lógica difusa al sistema experto hará que las reglas, junto a los resultados, sean bastante mas precisos. Esto hará que el usuario final tenga mas claro como de certera es una acción.
- **Ampliar reglas:** Se han dejado numerosos movimientos, técnicas y tácticas fuera del sistema experto, es por esto que habrá que seguir trabajando con el experto, buscando nuevos a ser posible para tener diferentes puntos de vista, para cubrir todo el arbol táctico de espada. Para ello se deberá seguir trabajando igual que se hizo anteriormente, con sistema de entrevistas, analizando los resultados anteriores y confirmando el conocimiento adquirido con el experto. Después se traspasará dicho conocimiento al sistema experto. Otro punto a tener en cuenta es que actualmente solo se está presentando el caso inicial de un asalto. Un horizonte nuevo a explorar en el desarrollo del sistema sería introducir las acciones y/o tácticas ya realizadas con el resultado que obtuvieron y que el sistema sea capaz de recalcular los resultados en función de estos datos introducidos.

6.2.3. API predicción

- **Mejorar fiabilidad predicción:** El sistema de predicción no obtuvo un porcentaje de fiabilidad muy alto, es por esto por lo que se deberá trabajar en mejorarla. Para ello el primer paso será ampliar la BBDD. Lo primero y mas sencillo será obtener mas registros. Esto es fácil de realizar puesto que la base de obtención de datos ya está hecha, solo habría que renovar el código puesto que es posible que la página haya cambiado y ya no sirva, pero se juega con la carta de que hay varias competiciones al mes entre las diferentes categorías. Es por esto que será fácil ampliarla. Otro objetivo a conseguir será ampliar las columnas de cada registro. Se puede

intentar conseguir mas características de los tiradores como la altura de estos, categoría (no es lo mismo que se enfrenten dos de la misma, que un junior y un senior, aunque esto se subsana en parte con la edad), ranking obtenido en poules puesto que esto marca como estaba el tirador en ese día, etc. Con todo esto podremos mejorar el modelo de entrenamiento y con esto aumentaremos la fiabilidad del sistema.

- **Mejorar tiempos API calcular predicción:** Actualmente la API tarda entre tres y cuatro segundos (suponiendo que la máquina esté levantada y a pleno funcionamiento) lo cual hace que no podamos incluir esto en toda la parte de la aplicación. El método que corresponde con la llamada en la API junta el entrenamiento del modelo junto al cálculo de predicción en la llamada. Lo que mas tarda con diferencia, llegando a ser del 95 % del tiempo es el entrenamiento del modelo. Es por esto que si se separara el entrenamiento del modelo, que con hacerse una vez debería ser suficiente, del cálculo de predicción se debería ahorrar mucho tiempo en la llamada de predicción. El resultado de este cambio debería ser dejar la respuesta de dicho endpoint prácticamente inmediata

6.2.4. Aplicación web

- **Mejorar UX web:** Para reducir los tiempos que tarda el usuario y mejorar la comprensión de estos se deberá rehacer los formularios o mejorar ciertos aspectos como los seleccionables siendo estos mas específicos. También se deberá añadir una ayuda rápida y explicativa de que es cada campo y para que sirve. Por otro lado habrá que revisar la distribución de los botones para ahorrar tiempo y que sean mas fáciles de encontrar cada sección.
- **Mejorar mensajes feedback resultados:** Los resultados son poco explicativos debido a que solo te indican si debes realizar una acción o no. Esto es útil para aquellos usuarios que tengan cierta experiencia y puedan discernir que es dicha acción. En un futuro se deberá ampliar estos mensajes, indicando que es dicha acción o táctica y porque se ha tomado dicha decisión.
- **Sección FAQ:** Prácticamente toda web con este formato tiene un apartado de preguntas frecuentes el cuál no ha sido desarrollado por falta de tiempo. Incluir este apartado puede ayudar positivamente a aquellos usuarios que tengan ciertas dudas sobre el sistema o incluso con el mundo de la esgrima.
- **Foro:** Incluir un foro puede ser muy enriquecedor para la aplicación debido a que los usuarios pueden compartir diferentes opiniones sobre los resultados y ayudarse entre ellos. Esto abre la posibilidad a generar una comunidad alrededor de la aplicación, haciendo de esta un factor mas a tener en cuenta en el mundo de la esgrima.
- **Usuarios:** Añadiendo usuarios se podrá habilitar y mejorar la posibilidad del foro mencionada anteriormente. También abre otras tantas puertas como la gestión de perfiles de usuario guardando el idioma por defecto, sin tener que cambiarlo cada vez que acceda a la aplicación. También se podrán guardar perfiles de tiradores, pudiendo hacer las consultas al sistema cargando dichos tiradores en vez introducir sus características, agilizando de este modo el proceso. Asimismo se podrá guardar el historial de las consultas realizadas, ahorrando consultas, evitando cargas extra al sistema y ahorrando tiempo al usuario.
- **Adaptación responsive:** La aplicación solo fue desarrollada pensando en el uso desde el ordenador de escritorio. Es por esto que se deberá seguir trabajando en la adaptación a diferentes dispositivos como los móviles, tablets, etc. También se deberá facilitar el uso para personas con cierta discapacidad visual, adaptando la web a dicha para que el uso de esta para dichas personas sea lo mas cómodo posible.
- **Mejora SEO:** Dedicando recursos al SEO conseguiremos que la aplicación web sea mas fácil de encontrar para los usuarios, haciendo que sea mas conocida la página y por tanto mas utilizada.

6.2.5. General

- **Sistema experto para entrenamiento:** Durante el proceso de desarrollo de la aplicación y tras recibir feedback de los usuarios que la utilizaron en las pruebas se llegó a la conclusión de que un apartado de entrenamiento aportaría mucho a la aplicación. Es obvio que esto es algo totalmente diferente al sistema de apoyo a la toma de decisión, pero si se complementan debido a que esta parte puede ayudar mucho a los clubes pequeños a perfeccionar y ampliar sus entrenamientos. Se hizo un primer desarrollo indicando acciones de manera aleatoria para trabajar durante los asaltos de entrenamiento. Pero el objetivo principal de esto es desarrollar un sistema experto el cuál nos indique un entrenamiento a realizar en función de nuestras fortalezas y debilidades, nuestras características como tiradores (altura, puño utilizado, estilo de esgrima, acciones conocidas, etc), tiempo disponible y forma física de la que se parte.
- **Cambiar a plan pago para mejorar tiempos:** Una de las limitaciones impuestas por el plan gratuito de Heroku son los tiempos de inicio de los servidores siendo estos de unos 25-30 segundos. Ampliando el plan a uno de pago, siendo este de 7 dolares al mes, evitaremos que se apaguen los servidores. También podremos comprar un dominio par que el acceso a este dando una mejor imagen de la página. Para esto se podrá buscar algún tipo de financiación como la publicidad dentro de la aplicación o algún tipo de subvencion de las diferentes federaciones, tanto regionales como la española, incluso llegando a presentar a la internacional el proyecto.
- **Añadir sección florete y sable:** Como se menciono antes el sistema solo está pensado para espada, faltando florete y sable. Añadiendo estas dos armas conseguiremos cubrir todo el público. Para ello deberemos seguir todo el proceso llevado a cabo con la espada, desarrollando un sistema experto basado en conocimiento, extrayendo este de un experto en la materia.
- **Crear aplicación móvil offline:** Uno de los problemas que se mencionó antes fue la falta de acceso a la aplicación cuando no se tiene acceso a internet, además de la dificultad de uso desde el móvil. Por estas razones se deberá desarrollar una aplicación móvil, tanto android como ios. Del mismo modo también será conveniente desarrollar una aplicación de escritorio para aquellos usuarios que utilicen un PC en sus lugares de entrenamiento

ANEXO A

TEST DE SLAGEL

pendiente escribir

ENTREVISTAS

Fecha:

Hora:

Lugar:

Asistentes:

Conocimiento anterior a la entrevista: Síntesis del conocimiento obtenido de las entrevistas anteriores.

Objetivos de la entrevista: El objetivo que se pretende alcanzar con la entrevista.

Fuentes de conocimiento: Personas de las cuales se obtiene el conocimiento o lo que es lo mismo, las personas que van a ser entrevistadas.

Modo: Entrevista estructurada o parcialmente estructurada. Para identificación de dichos errores.

Planteamiento de la sesión: En este apartado se muestran las preguntas que se desean realizar para obtener el conocimiento.

Resultado de la sesión: Aquí se transcriben las respuestas obtenidas a las preguntas del planteamiento de la sesión.

Plan de análisis: Pasos a realizar para analizar.

Resultado del análisis: Resultado final de la entrevista.

A continuación se muestran algunas entrevistas realizadas a partir del formato anterior.

B.1. ENTREVISTA 1

Fecha: Jueves 20 de Abril de 2017.

Hora: 19:40

Lugar: Sala de armas Espada de Calatrava

Asistentes:

- Juan Lomas Rayego (Experto).
- Gregorio B. Patiño Esteo (IC).

Situación del análisis respecto al modelo general: Esta entrevista es la primera a realizar dentro del conjunto de entrevistas y diferentes técnicas para la adquisición del conocimiento necesario para realizar el prototipo de sistema experto (S.E). Esta será la primera entrevista por la que haremos preguntas muy generales y sobre la marcha iremos haciendo preguntas sobre las que tengamos dudas.

Conocimiento anterior a la entrevista: Puesto que es la primera de las entrevistas el conocimiento anterior es nulo, por tanto no tendremos conocimiento previo exceptuando el adquirido por la investigación previa, que es el sistema de puntaje y las normas.

Objetivos de la entrevista:

- (A) Identificar Las características en las que hay que fijarse en ambos tiradores para determinar la táctica.
- (B) Determinar las relaciones en estas características para saber la táctica a seguir.

Fuentes de conocimiento: Experto. La razón por la que se llevó a cabo esta elección es que además de ser entrenador es tirador y tiene bastantes años a sus espaldas que le respaldan, además de haber participado en el circuito nacional y haber asistido a varias concentraciones de la selección española como técnico. Por tanto ambos objetivos podrían cumplirse ya que con su experiencia como tirador podría identificar las características en las que habría que fijarse (objetivo A) y su conocimiento teórico adquirido como entrenador (con sus respectivos cursos y exámenes) podría relacionarlos para saber qué hacer en cada caso (objetivo B).

Modo: El modo en el que se hará la entrevista será abierta. Realizando preguntas bastante abiertas para que el experto pueda expresarse libremente y de esta manera adquirir el mayor glosario posible y además poder conocer cuáles pueden ser los casos. En caso de que el conocimiento sea adquirido de una buena forma podría empezarse a preguntar cosas más específicas sobre ciertas situaciones.

Planteamiento de la sesión: En este apartado se muestran las preguntas que se desean realizar para obtener el conocimiento.

- A1. ¿Qué es lo primero en lo que te fijas a la hora de planificar una táctica?
- A2. ¿Cómo determinas si tiene más distancia que tú?
- A3. ¿Cómo actúa la experiencia a la hora de determinar la táctica?
- A4. ¿Qué puedes decirme sobre el físico?
- B1. ¿Si tiene más distancia que tu como lo valoras?
- B2. ¿Qué puedes decirme sobre los puños franceses y anatómicos, ventajas y cómo actuar?
- B3. ¿Echar de la pista o ceder terreno?
- B4. ¿Ser agresivo o pasivo?

Resultado de la sesión: Aquí se transcriben las respuestas obtenidas a las preguntas del planteamiento de la sesión.

A1. ¿Qué es lo primero en lo que te fijas a la hora de planificar una táctica?
Si tiene más distancia, experiencia y físico.

B1. ¿Si tiene más distancia que tu como lo valoras?

Lo que hago es intentar usar técnicas que me permitan entra una técnica en una distancia en la que yo puedo tocar y el no, entonces lo que necesito es entrar con hierro o con segunda intención.

A2. ¿Cómo determinas si tiene más distancia que tú?

Por la altura suele ser un buen indicador otro es si lleva francesa y yo anatómica y luego haces una valoración.

A3 ¿Cómo actúa la experiencia a la hora de determinar la táctica?

Si es más joven que yo es una persona que no sabe mucho y se siente intimidado por mí por lo que intento ganarle más rápidamente, o si esta echado hacia delante lo espero, provoco que cometa errores y le gano en contra.

A4 ¿Qué puedes decirme sobre el físico? Si tiene más físico que yo, por ejemplo no puedo pretender ser más rápido que una persona que es más rápida que yo, entonces la técnica cambia, obviamente si él es más lento que yo quizás siendo más ofensivo es buena opción, si es más rápido que yo tengo que esperarme más o jugar con una segunda intención.

B2 ¿Qué puedes decirme sobre los puños franceses y anatómicos, ventajas y cómo actuar? Puño francés sus ventajas son alcance y poco más, actuaría con esgrima de poco contacto en hierro evitando el contacto con el rival y una distancia larga media siempre, favorece un mejor físico Anatómica pierdes distancia pero ganas puntería y tienes que entrar en una distancia media baja y necesita más técnica.

B3 ¿Echar de la pista o ceder terreno? Si veo que la otra persona no ataca, le llevo a final de pista y no puede moverse o le toco o se sale. Si me vuelvo loco y es una persona que para muy bien o que al contrataque va muy bien seguramente me meta muchos puntos, sin embargo, si tiene una esgrima de mucho más ímpetu dejo que ataque, le dejo corto y le doy después.

B4 ¿Ser agresivo o pasivo? Normalmente suelo ser más precavido por mi personalidad, pero hay veces que si eres agresivo puedes intimidar al otro como cuando tienen menos experiencia que tu o es más joven que tú, a esa gente hay que ganarle rápido antes de que pueda pensar nada. Que es alguien con más ímpetu que tú, dejas que se precipite y le ganas entonces eres más pasivo

Plan de análisis: Pasos a realizar para analizar.

- Identificar términos usados en la entrevista
- Identificar características tirador
- Comprender importancia y relación de las características de un tirador
- Aumentar glosario

Resultado del análisis: Resultado final de la entrevista. Términos:

- Distancia
- Experiencia
- Físico
- Altura
- Puño
- Francés
- Anatómico

- Técnica
- Agresividad
- Parada

Características de un tirador:

- Altura
- Físico
- Experiencia
- Juventud
- Agresivo

Relación características tirador:

Después de la entrevista se ha comprendido que el estado físico de una persona cobra importancia en la esgrima. Siendo este un factor importante pero no determinante, el cual se puede compensar de otras maneras, ya sea con experiencia u otras técnicas. Quizás la característica que destaca mas sobre el resto sería la experiencia, puesto que esta es la que te permitirá adaptarte dentro del asalto. De todas formas esto se afianzará en próximas entrevistas.

B.2. ENTREVISTA 2

Fecha: Viernes 4 de Mayo de 2017.

Hora: 19:40

Lugar: Sala de armas Espada de Calatrava

Asistentes:

- Juan Lomas Rayego (Experto).
- Gregorio B. Patiño Esteo (IC).

Situación del análisis respecto al modelo general: Con la anterior entrevista finalizada y resueltas todas las dudas mediante medios de comunicación digitales para favorecer la fluidez de estos mismos, estamos mas cerca de alcanzar el modelo general. Es por esto por lo que esta segunda entrevista será dedicada a ampliar los conocimientos adquiridos consiguiendo entrar en movimientos no tan básicos y combinaciones más complejas

Conocimiento anterior a la entrevista: Partimos de una base de conocimiento adquirido por la anterior entrevista, en la cual se vieron características básicas. Este junto al anterior obtenido mediante investigación propia ya nos da una licencia para entender algo sobre tácticas en esgrima. También podremos formar nuestra propia táctica, pero esta sera muy básica. El conocimiento que tenemos son los movimientos básicos junto a las características principales.

Objetivos de la entrevista:

- (A) Ampliar base movimientos.
- (B) Ampliar base combinaciones.

(C) Ampliar base características.

Fuentes de conocimiento: Experto. La razón por la que se llevó a cabo esta elección es que además de ser entrenador es tirador y tiene bastantes años a sus espaldas que le respaldan, además de haber participado en el circuito nacional y haber asistido a varias concentraciones de la selección española como técnico. Por tanto todos los objetivos podrían cumplirse ya que con su experiencia como tirador podría identificar las características en las que habría que fijarse (objetivo C) y su conocimiento teórico adquirido como entrenador (con sus respectivos cursos y exámenes) podría relacionarlos para saber qué hacer en cada caso (objetivo A y B).

Modo: El modo en el que se hará esta entrevista será semi-abierta. Algunas de las preguntas serán con respuesta totalmente abiertas, mientras que otras de ellas serán propuestas que haremos nosotros para saber si vamos cogiendo el conocimiento de una manera correcta, y en caso de que no lo fuera corregirlo.

Planteamiento de la sesión: En este apartado se muestran las preguntas que se desean realizar para obtener el conocimiento.

- A1. ¿Hay alguna manera de saltarse la guardia del rival?
- A2. ¿Hay alguna manera de forzar al rival para que ataque?
- A3. ¿Hay alguna manera de distraer al rival legalmente?
- B1. ¿Hay alguna manera de engañar al rival con segundas intenciones?
- B2. ¿Hay alguna manera de engañar al rival en la distancia?
- C1. ¿Como de importante es que el tirador sepa hacer bien las paradas?
- C2. ¿Como de importante es que el tirador sepa atacar bien?
- C3. ¿Como de importante es que el tirador caiga en las trampas y engaños?
- C4. ¿Como influye el paso del tiempo en el asalto?
- C5. ¿Como actúa la confianza en el tirador dentro del asalto?
- C6. ¿Como influye la confianza en el tirador a la hora de planificar la táctica?

Resultado de la sesión: Aquí se transcriben las respuestas obtenidas a las preguntas del planteamiento de la sesión.

A1. ¿Hay alguna manera de saltarse la guardia del rival?

Sí, existe un movimiento llamado lanzado o mas conocido como coupé el cual te permite saltar la guardia del rival. La única defensa ante este ataque es quitar el brazo o intentar tocar con un contra-ataque al rival para que tu luz se encienda antes que la del otro, asumiendo los riesgos de fallar o no tocar con suficiente antelación.

A2. ¿Hay alguna manera de forzar al rival para que ataque? Sí. Para ello están las fintas. Las fintas es un movimiento por el cual tu engañas a tu rival para provocar la acción que tu quieras. Por ejemplo, si yo quiero hacer algún movimiento, pero para este necesito que intentes tocarme en el pie. En este caso esperar a que el rival vaya al pie puede ser muy largo o no darse el caso. Por esto mismo podríamos provocar al rival dejando el pie al descubierto para que este tenga mas ganas de ir.

A3. ¿Hay alguna manera de distraer al rival legalmente? Sí. Toda acción que hagas es legal siempre y cuando no intimides al rival, no pongas en riesgo tu seguridad ni sea antideportivo. Por ejemplo, ponerte a gritar podría ser algo antideportivo además de intimidar al rival, por tanto eso no lo podrías hacer. Algo que si podrías hacer sería dar pisotones de vez en cuando en el suelo. Esto hará que tu rival se distraiga y entonces tu aprovechar para lanzar un ataque. Hay gente que esto no lo considera del todo honorable pero eso ya a criterio de cada uno. Algo con mas de estilo es dar golpes con tu espada en la suya, suaves y fuertes, variando entre ellos. De esta manera tu rival estará mas pendiente de tu espada y podrás aprovechar para disumular otras acciones con el cuerpo, como ganarle distancia.

B1. ¿Hay alguna manera de engañar al rival con segundas intenciones?

Sí. Al igual que hablabamos antes que puedes hacer acciones con el cuerpo para distraer al rival, también las puedes hacer con la propia espada a modo de acciones. Puedes hacer un movimiento amagando que vas a atacar a un punto suyo en concreto y cuando el vaya a defenderse, entonces cambiar el objetivo y atacar a otro sitio. Esto es conocido como 1-2 o mas técnicamente finta-pase.

B2. ¿Hay alguna manera de engañar al rival en la distancia?

Sí. Con la propia guardia puedes engañarle, si la avanzas el rival se pensará que estás más cerca de lo que realmente te encuentras ya que tu punta se encontrará mas cerca de él. Por otro lado si la retrasas conseguirás que piense que estás mas lejos, pudiendo aprovechar esto para tocar con mayor facilidad en tocados a corta distancia. Por otro lado también se puede hacer con un movimiento constante de piernas en el que la distancia de este no sea siempre la misma.

C1. ¿Como de importante es que el tirador sepa hacer bien las paradas?

Es tan importante como que esto determinará si podemos ser mas agresivos o no en nuestros ataques. Incluso siendo algo extremistas, podría ser lo que indique si debemos atacar durante el asalto. Ante un oponente que ejecute a la perfección sus paradas y no tenga huecos en la defensa, lo mejor no será atacar puesto que esto provocará en la mayoría de los casos que cometamos errores atacando y de ese modo obtendremos puntos en contra, y eso nunca lo queremos. Por el contrario, si el rival tiene muchos huecos en defensa nos centraremos en atacar.

C2. ¿Como de importante es que el tirador sepa atacar bien?

Es lo mismo que hemos hablado antes solo que se invertirían los roles entre tiradores. No puedes intentar defender si el rival es muy superior a ti en su ataque.

C3. ¿Como de importante es que el tirador caiga en las trampas y engaños?

Otro factor que dirá si debemos hacer ataques indirectos o directos. Si el rival siempre cae en nuestros engaños lo mejor será elaborar mas el tocado para asegurarlo. Por el contrario si el rival nunca cae en nuestros engaños lo mejor será centrarnos en la explosividad e ir lo mas recto posible.

C4. ¿Como influye el paso del tiempo en el asalto?

Esto influirá en como de agresivos han de ser los tiradores en función del resultado. En caso de que vayas perdiendo y el resultado sea muy distante, el tirador que pierda deberá intentar dejar pasar el menor tiempo posible, puesto que este juega en su contra. Por otro lado, el que gane podrá permitirse la licencia de dejar correr el cronometro. Esto es debido a que cuanto mas tiempo pase estando el por encima, ma

C5. ¿Como actua la confianza en el tirador dentro del asalto?

Esto es bastante difícil de determinar. Cuando un tirador esté repleto de confianza intentará tomar decisiones más arriesgadas, mientras que cuando esté bajo de esta lo más probable es que intente actuar bajo seguro esperando los errores del rival. Por tanto este podría ser un indicativo sobre la confianza que tiene el rival en él. Pero de nuevo, insisto en que cada persona actúa de una forma distinta ante la ausencia o exceso de confianza.

C6. ¿Cómo influye la confianza en el tirador a la hora de planificar la táctica?

Bien esto es algo complejo. Habrá que distinguir por dos partes. En cuanto a nuestro tirador hay que tener cuidado con el exceso de confianza puesto que esto nos hará cometer errores. Por otro lado hay que tener cuidado también con la falta de confianza, puesto que esto hará que dudemos y perderemos tiempo en las acciones, lo que provocará que sea más fácil que nos toquen.

Plan de análisis: Pasos a realizar para analizar.

- Identificar términos usados en la entrevista
- Identificar características nuevas tirador
- Comprender importancia y relación de las nuevas características de un tirador
- Aumentar glosario

Resultado del análisis: Resultado final de la entrevista. Términos:

- Coupé
- Guardia
- Defensa
- Contra-ataque
- Finta
- Provocar
- Engañar
- Distraer
- Llamada
- Finta-Pase

Características de un tirador:

- Reflejos
- Capacidad defensiva
- Capacidad ofensiva
- Confianza
- Explosividad

Relación nuevas características tirador:

Se habló sobre los reflejos y la explosividad, por lo que se entiende que la velocidad en un tirador es importante. Esto hace que haya que tener realmente en cuenta. Por otro lado también se habló de las capacidades del tirador, tanto ofensivas como defensivas. Esto será importante a la hora de decantarnos por una táctica u otra. También se mencionó la confianza que tiene un tirador en si mismo. Esto se puede aprovechar tanto en el rival como en nosotros. Detectando si tiene exceso o escasez de esta podremos ser mas agresivos o tendremos que ser mas conservadores.

DIAGRAMAS TOMA DECISIÓN

En este anexo se pueden encontrar los diagramas de toma de decisión desarrollados para el sistema experto. Se muestran a continuación.



Figura C.1: Esquema toma decisión

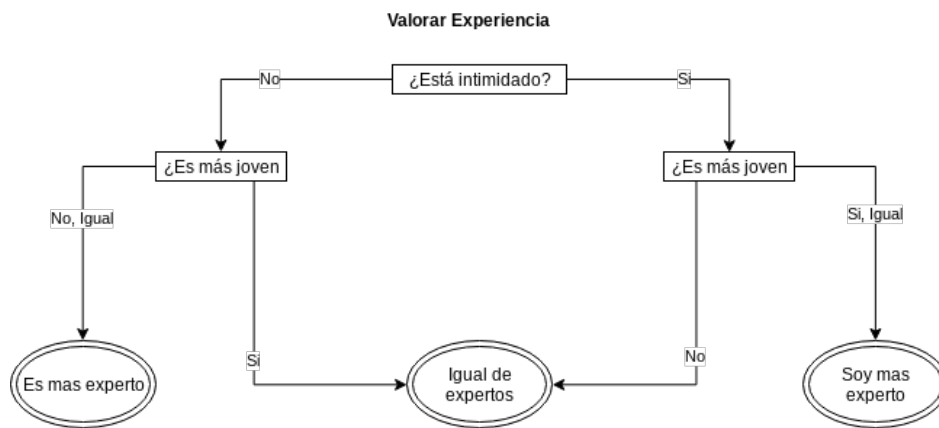


Figura C.2: Árbol decisión experiencia

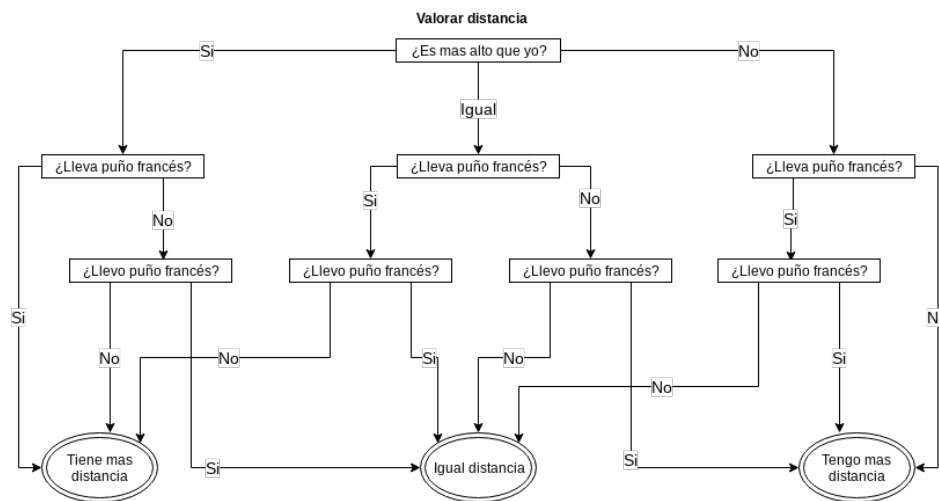


Figura C.3: Árbol decisión distancia

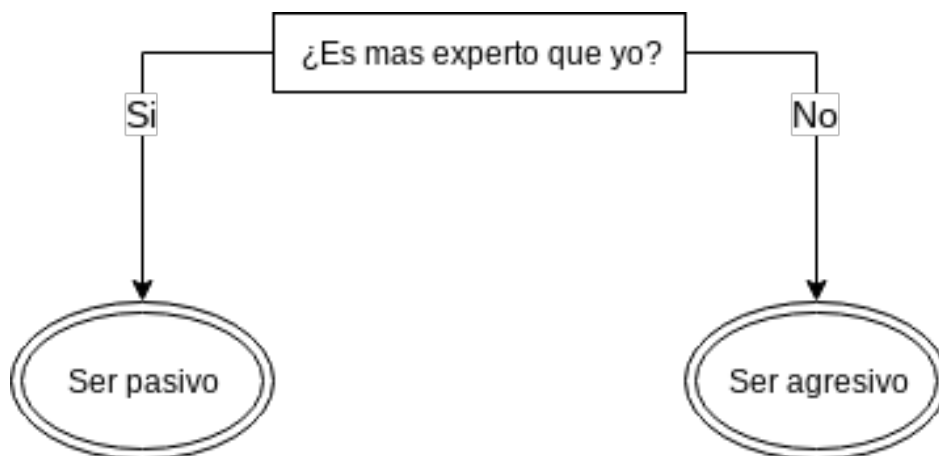


Figura C.4: Árbol decisión agresividad



Figura C.5: Árbol decisión acortar distancia

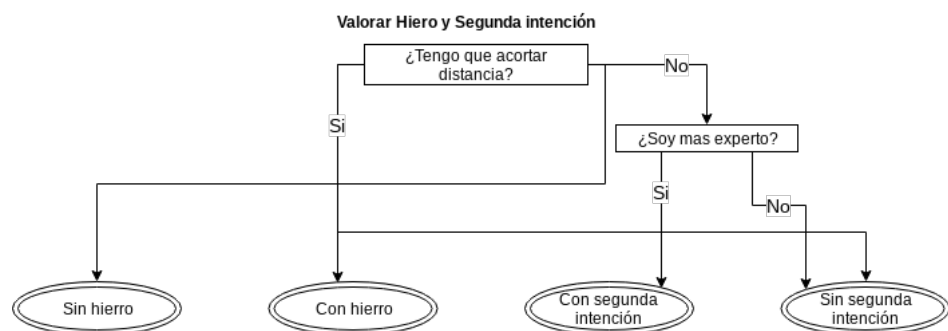


Figura C.6: Árbol decisión hierro y segunda intención

REGLAS DEL SISTEMA EXPERTO

A continuación se lista la plantilla y reglas utilizadas para el sistema experto en CLIPS.

```

1  ;-----
2  ;
3  ;     PLANTILLA
4  ;
5  ;-----
6
7
8
9  (deftemplate tirador
10 (slot id
11   (type SYMBOL)
12   (allowed-symbols yo el)
13   (default el)
14 )
15 (slot edad
16   (type INTEGER)
17 )
18 (slot altura
19   (type INTEGER)
20 )
21 (slot puño
22   (type SYMBOL)
23   (allowed-symbols frances anatomico)
24   (default frances)
25 )
26 (slot intimidado
27   (type SYMBOL)
28   (allowed-symbols si no)
29   (default no)
30 )
31 )
32 )
33
34 ;-----
35 ;
36 ;     REGLAS
37 ;
38 ;-----
39
40 (defrule cogerDatosTiradorEl "coger datos"
41
42   ?el <- (tirador (id el) (edad ?edadEl)(altura ?alturaEl)(puño ↵
43     ↵ ?puñoEl)(intimidado ?intimidadoEl))
44   ?yo <- (tirador (id yo) (edad ?edadYo)(altura ?alturaYo)(puño ↵
45     ↵ ?puñoYo))
46   ;(test (> ?edadEl ?edadYo))
47   =>

```

```

46     (assert
47         (alturaEl ?alturaEl)(alturaYo ?alturaYo)
48         (edadEl ?edadEl)(edadYo ?edadYo)
49         (puñoEl ?puñoEl)(puñoYo ?puñoYo)
50         (intimidadoEl ?intimidadoEl)
51     )
52
53     (retract ?el)
54     (retract ?yo)
55 )
56
57
58 ;-----
59 ;
60 ;         Sacamos la altura
61 ;
62 ;-----
63
64
65
66
67 (defrule compararAltura1
68     (alturaEl ?x)(alturaYo ?y)
69     (test (> ?x ?y))
70     (test (> (- ?x ?y) 5)) ;seria para comprobar que la ←
71         ↪ diferencia es mayor que 5
72     =>
73     (assert (altura masEl))
74 )
75
76 (defrule compararAltura1_2
77     (alturaEl ?x)(alturaYo ?y)
78     (test (> ?x ?y))
79     (test (<= (- ?x ?y) 5)) ;seria para comprobar que la ←
80         ↪ diferencia es mayor que 5
81     =>
82     (assert (altura igual))
83 )
84
85 (defrule compararAltura2_1
86     (alturaEl ?x)(alturaYo ?y)
87     (test (< ?x ?y))
88     =>
89     (assert (altura masYo))
90 )
91
92 (defrule compararAltura2_2
93     (alturaEl ?x)(alturaYo ?y)
94     (test (<= ?x ?y))
95     (test (<= (- ?y ?x) 5)) ;seria para comprobar que la ←
96         ↪ diferencia es mayor que 5
97     =>
98     (assert (altura igual))
99 )
100
101 ;-----
102 ;
103 ;         Sacamos la distancia
104 ;
105 ;-----

```

```
106
107 (defrule distancia_1
108   (altura masEl)
109   (puñoEl frances)
110   (puñoYo frances)
111   =>
112   (assert (distancia masEl))
113 )
114
115 (defrule distancia_2
116   (altura masEl)
117   (puñoEl anatomico)
118   (puñoYo anatomico)
119   =>
120   (assert (distancia masEl))
121 )
122
123 (defrule distancia_3
124   (altura masEl)
125   (puñoEl frances)
126   (puñoYo anatomico)
127   =>
128   (assert (distancia masEl))
129 )
130
131 (defrule distancia_4
132   (altura masEl)
133   (puñoEl anatomico)
134   (puñoYo frances)
135   =>
136   (assert (distancia igual))
137 )
138
139 (defrule distancia_5
140   (altura igual)
141   (puñoEl frances)
142   (puñoYo frances)
143   =>
144   (assert (distancia igual))
145 )
146
147 (defrule distancia_6
148   (altura igual)
149   (puñoEl anatomico)
150   (puñoYo anatomico)
151   =>
152   (assert (distancia igual))
153 )
154
155 (defrule distancia_7
156   (altura igual)
157   (puñoEl frances)
158   (puñoYo anatomico)
159   =>
160   (assert (distancia masEl))
161 )
162
163 (defrule distancia_8
164   (altura igual)
165   (puñoEl frances)
166   (puñoYo frances)
167   =>
168   (assert (distancia igual))
```

```

169 )
170 (defrule distancia_8_2
171   (altura igual)
172   (punioEl anatomico)
173   (punioYo frances)
174 =>
175   (assert (distancia igual))
176 )
177
178 (defrule distancia_9
179   (altura masYo)
180   (puñoEl frances)
181   (puñoYo frances)
182 =>
183   (assert (distancia masYo))
184 )
185
186 (defrule distancia_10
187   (altura masYo)
188   (puñoEl anatomico)
189   (puñoYo anatomico)
190 =>
191   (assert (distancia masYo))
192 )
193
194 (defrule distancia_11
195   (altura masYo)
196   (puñoEl frances)
197   (puñoYo anatomico)
198 =>
199   (assert (distancia igual))
200 )
201
202 (defrule distancia_12
203   (altura masYo)
204   (puñoEl anatomico)
205   (puñoYo frances)
206 =>
207   (assert (distancia masYo))
208 )
209
210 ;-----
211 ;
212 ;       Sacamos la joven
213 ;
214 ;-----
215
216 (defrule compararEdad1
217   (edadEl ?x)(edadYo ?y)
218   (test (= ?x ?y))
219   =>
220   (assert (joven igual))
221 )
222
223 (defrule compararEdad2
224   (edadEl ?x)(edadYo ?y)
225   (test (> ?x ?y))
226   =>
227   (assert (joven mayorEl))
228 )
229
230 (defrule compararEdad3
231   (edadEl ?x)(edadYo ?y)

```

```

232 (test (< ?x ?y))
233 =>
234 (assert (joven mayorYo))
235 )
236
237
238
239 ;-----
240 ;
241 ;         Sacamos la experiencia
242 ;
243 ;-----
244
245 (defrule experimentado_1
246   (intimidadoEl si)
247   (joven mayorEl)
248   =>
249   (assert (experiencia igual))
250 )
251
252 (defrule experimentado_2
253   (intimidadoEl si)
254   (joven igual)
255   =>
256   (assert (experiencia masYo))
257 )
258
259 (defrule experimentado_3
260   (intimidadoEl si)
261   (joven mayorYo)
262   =>
263   (assert (experiencia masYo))
264 )
265
266 (defrule experimentado_4
267   (intimidadoEl no)
268   (joven mayorEl)
269   =>
270   (assert (experiencia masEl))
271 )
272
273 (defrule experimentado_5
274   (intimidadoEl no)
275   (joven igual)
276   =>
277   (assert (experiencia igual))
278 )
279
280 (defrule experimentado_6
281   (intimidadoEl no)
282   (joven mayorYo)
283   =>
284   (assert (experiencia masYo))
285 )
286
287 ;-----
288 ;
289 ;         Sacamos la agresividad
290 ;
291 ;-----
292
293
294 (defrule agresividad_1

```

```

295 (or (experiencia masEl)(experiencia igual))
296 =>
297 (assert (agresividad no))
298 )
299
300
301 (defrule agresividad_2
302 (experiencia masYo)
303 =>
304 (assert (agresividad si))
305 )
306
307
308 ;-----
309 ;
310 ;         Sacamos la acortar distancia
311 ;
312 ;-----
313
314 (defrule acortarDistancia_1
315 (distancia masEl)
316 (agresividad si)
317 =>
318 (assert (acortarDistancia si))
319 )
320
321 (defrule acortarDistancia_2
322 (or(distancia masYo) (distancia igual))
323 (agresividad no)
324 =>
325 (assert (acortarDistancia no))
326 )
327
328 (defrule acortarDistancia_3
329 (or(distancia masYo) (distancia igual))
330 (agresividad si)
331 =>
332 (assert (acortarDistancia no))
333 )
334
335 (defrule acortarDistancia_4
336 (distancia masEl)
337 (agresividad no)
338 =>
339 (assert (acortarDistancia no))
340 )
341
342
343 ;-----
344 ;
345 ;         Sacamos la contacto hierro
346 ;
347 ;-----
348
349 (defrule contactoHierro_1
350 (acortarDistancia si)
351 =>
352 (assert (contactoHierro si))
353 )
354
355 (defrule contactoHierro_2
356 (acortarDistancia no)
357 =>

```



```
358 (assert (contactoHierro no))
359 )
360
361 ;-----
362 ;
363 ;     Sacamos segunda intencion
364 ;
365 ;-----
366
367 (defrule segundaIntencion_1
368   (acortarDistancia si)
369   (experiencia masEl)
370   =>
371   (assert (segundaIntencion no))
372 )
373
374 (defrule segundaIntencion_2
375   (acortarDistancia no)
376   (experiencia masYo )
377   =>
378   (assert (segundaIntencion si))
379 )
380
381 (defrule segundaIntencion_3
382   (acortarDistancia si)
383   (or (experiencia masYo)(experiencia igual))
384   =>
385   (assert (segundaIntencion no))
386 )
387
388 (defrule segundaIntencion_4
389   (acortarDistancia no)
390   (or(experiencia masEl)(experiencia igual))
391   =>
392   (assert (segundaIntencion no))
393 )
394
395
396
397
398 (defrule imprimir
399   (contactoHierro ?x)
400   (segundaIntencion ?y)
401   =>
402   (printout t "¿Contacto hierro? " ?x crlf)
403   (printout t "¿Segunda intencion ? " ?y crlf)
404 )
```


ALGORITMO FUERZA BRUTA

OBTENCIÓN MEJORES VARIABLES

KNN

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  #Pandas
5  import pandas as pd
6  import numpy as np
7
8  #Cv
9  from sklearn.cross_validation import train_test_split
10
11 #Classifier imports
12 from sklearn.neighbors import KNeighborsClassifier
13 from sklearn.metrics import auc, accuracy_score, confusion_matrix
14 from sklearn import metrics
15
16 #Plot libraries
17 import matplotlib.pyplot as plt
18
19 #Get time
20 import datetime
21 import time
22
23
24 aa = ["TABLEU", "C1_AGE", "C1_RANKING", "C1_HANDNESS", ↵
      ↵ "C1_WEAPON", "C2_AGE",
25      ↵ "C2_RANKING", "C2_HANDNESS", "C2_WEAPON", "WINNER"]
26
27 dataframe = pd.read_csv(r"6_competition_rankings_columns.csv")
28 dataframe = pd.read_csv(r"6_competition_rankings_columns.csv", ↵
      ↵ header=1, names=aa)
29 dataframe = dataframe.drop(dataframe[dataframe['C2_WEAPON'] == ↵
      ↵ ' None '].index)
30 dataframe = dataframe.drop(dataframe[dataframe['C1_WEAPON'] == ↵
      ↵ ' None '].index)
31 dataframe.to_csv("testing_csv")
32
33
34 X = np.array(dataframe.drop(['WINNER'],1))
35 y = np.array(dataframe['WINNER'])
36 x1, x2, y1, y2 = train_test_split(X, y, random_state = 100)
37
38
39
```

```

40
41 '''
42 The next code got a large computational time. Be sure to have a ↵
43     ↵ film to watch
44 while it execute
45 '''
46
47 print("start of script")
48 print(datetime.datetime.now())
49 start_time = datetime.datetime.now()
50
51 # Create Dataframe from our data.
52 aa = ["TABLEU", "C1_AGE", "C1_RANKING", "C1_HANDNESS", ↵
53     ↵ "C1_WEAPON", "C2_AGE",
54     "C2_RANKING", "C2_HANDNESS", "C2_WEAPON", "WINNER"]
55
56 dataframe = pd.read_csv(r"6_competition_rankings_columns.csv")
57 dataframe = pd.read_csv(r"6_competition_rankings_columns.csv", ↵
58     ↵ header=1, names=aa)
59 dataframe = dataframe.drop(dataframe[dataframe['C2_WEAPON'] == ↵
60     ↵ ' None '].index)
61 dataframe = dataframe.drop(dataframe[dataframe['C1_WEAPON'] == ↵
62     ↵ ' None '].index)
63 dataframe.to_csv("testing_csv")
64
65 # Split data from train to test
66 X = np.array(dataframe.drop(['WINNER'],1))
67 y = np.array(dataframe['WINNER'])
68 x1, x2, y1, y2 = train_test_split( X, y, random_state = 100)
69 X_train, X_test, y_train, y_test = train_test_split(X, y, ↵
70     ↵ test_size=0.4, random_state=3)
71
72 # Initialize max variables
73 max_score = 0
74 max_k = 0
75 max_leaf_size = 0
76 max_p = 0
77
78 k_iterations = range(0, 10)
79 leaf_size_ranges = range(1, 60)
80 p_ranges = range(1, 20)
81
82 # Test different K value
83 for k_value in k_iterations:
84     k_range = range((k_value * 25) + 1, (k_value * 25) + 26)
85     scores = []
86     for k in k_range:
87
88         # Explore the value for leaf_sie
89         for leaf_size_i in leaf_size_ranges:
90
91             # Explore the value for p
92             for p_value in p_ranges:
93                 knn = KNeighborsClassifier(n_neighbors=k,
94                     leaf_size=leaf_size_i,
95                     p=p_value,
96                     n_jobs=5)
97
98                 knn.fit(X_train, y_train)
99                 y_pred = knn.predict(X_test)
100                 score = accuracy_score(y_test, y_pred)
101                 if score > max_score:

```

```
97         max_score = score
98         max_k = k
99         max_leaf_size = leaf_size_i
100        max_p = p_value
101
102        print("End of k test:")
103        print(k)
104
105        print("End of iteration:")
106        print(k_value)
107        print("Time elapsed:")
108        print(datetime.datetime.now() - start_time)
109
110
111    print("max_score")
112    print(max_score)
113    print("max_k")
114    print(max_k)
115    print("max_leaf_size")
116    print(max_leaf_size)
117    print("max_p")
118    print(max_p)
119
120    print("end of the script")
121    end_time = datetime.datetime.now()
122    print(end_time)
123    print("Total time:")
124    print(end_time - start_time)
```


ANEXO F

BOCETOS HERRAMIENTA WEB

En este anexo se pueden consultar los bocetos iniciales de la herramienta web.

MFTInicioLlenarGuiadoContactoSugerenciasEntrenamientoInglés

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Figura F.1: Boceto página inicio

MFTInicioLlenarGuiadoContactoSugerenciasEntrenamientoInglés

Formulario completo sistema

Tirador 1

Selecciona el puño

Francés

Anatómico

Marque la siguiente opción si está intimidado

Intimidado

Edad

5

Altura (cm)

5

Tirador 2

Selecciona el puño

Francés

Anatómico

Marque la siguiente opción si está intimidado

Intimidado

Edad

5

Altura (cm)

5

Calcular

Figura F.2: Boceto página llenar

MFT

Inicio

Llenar

Guiado

Contacto

Sugerencias

Entrenamiento

Inglés

Formulario guiado sistema

¿Tienes mas distancia?

☐ Si

☐ No

☐ Igual

¿Eres mas joven?

☐ Si

☐ No

☐ Igual

Marque la siguiente opción si está intimidado

☐ Intimidado

Calcular

Figura F.3: Boceto página guiado

MFT

Inicio

Llenar

Guiado

Contacto

Sugerencias

Entrenamiento

Inglés

Hierro: es conveniente que cojas hierro antes de entrar. Esto suele darse en las situaciones que tengas que acortar distancia y tengas mas fuerza que él.

Segunda intención: no es bueno que intentes hacer hacer ataques en segunda intención. Si esto ha salido negativo puedes intentar a tocar por velocidad. Suele salir negativo en los casos de que el otro sea mas experimentado.

Figura F.4: Boceto página resultado

MFT

Inicio

Llenar

Guiado

Contacto

Sugerencias

Entrenamiento

Inglés

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec eratLorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Figura F.5: Boceto página contacto



Figura F.6: Boceto página contacto

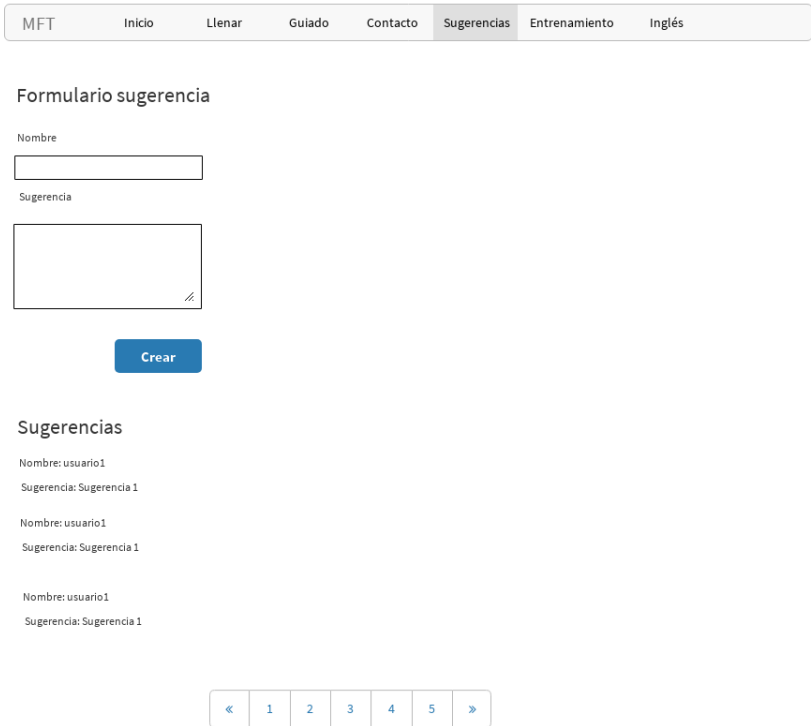


Figura F.7: Boceto página sugerencia

MFT

Inicio

Llenar

Guiado

Contacto

Sugerencias

Entrenamiento

Inglés

Generador acciones

Acciones

Mostrar

Numero acciones

5

Generar acciones

Figura F.8: Boceto página entrenamiento

ALGORITMO VALORES KNN

A continuación se lista el algoritmo para obtener los mejores valores del algoritmo KNN.

```
1 print ("start of script")
2 print(datetime.datetime.now())
3 start_time = datetime.datetime.now()
4
5
6 # Create Dataframe from our data.
7 aa = ["TABLEU", "C1_AGE", "C1_RANKING", "C1_HANDNESS", "
      ↪ "C1_WEAPON", "C2_AGE",
8       "C2_RANKING", "C2_HANDNESS", "C2_WEAPON", "WINNER"]
9
10 dataframe = pd.read_csv(r"6_competition_frankings_columns.csv")
11 dataframe = pd.read_csv(r"6_competition_frankings_columns.csv", ↪
      ↪ header=1, names=aa)
12 dataframe = dataframe.drop(dataframe[dataframe['C2_WEAPON'] == ↪
      ↪ ' None '].index)
13 dataframe = dataframe.drop(dataframe[dataframe['C1_WEAPON'] == ↪
      ↪ ' None '].index)
14 dataframe.to_csv("testing_csv")
15
16
17 # Split data from train to test
18 X = np.array(dataframe.drop(['WINNER'],1))
19 y = np.array(dataframe['WINNER'])
20 x1, x2, y1, y2 = train_test_split( X, y, random_state = 100)
21 X_train, X_test, y_train, y_test = train_test_split(X, y, ↪
      ↪ test_size=0.4, random_state=3)
22
23 # Initialize max variables
24 max_score = 0
25 max_k = 0
26 max_leaf_size = 0
27 max_p = 0
28
29 k_iterations = range(0, 10)
30 leaf_size_ranges = range(1, 60)
31 p_ranges = range(1, 20)
32
33 # Test different K value
34 for k_value in k_iterations:
35     k_range = range((k_value * 25) + 1, (k_value * 25) + 26)
36     scores = []
37     for k in k_range:
38
39         # Explore the value for leaf_sie
40         for leaf_size_i in leaf_size_ranges:
41
42             # Explore the value for p
```

```
43         for p_value in p_ranges:
44             knn = KNeighborsClassifier(n_neighbors=k,
45                                     leaf_size=leaf_size_i,
46                                     p=p_value,
47                                     n_jobs=5)
48             knn.fit(X_train, y_train)
49             y_pred = knn.predict(X_test)
50             score = accuracy_score(y_test, y_pred)
51             if score > max_score:
52                 max_score = score
53                 max_k = k
54                 max_leaf_size = leaf_size_i
55                 max_p = p_value
56
57         print ("End of k test:")
58         print (k)
59
60     print ("End of iteration:")
61     print (k_value)
62     print ("Time elapsed:")
63     print (datetime.datetime.now() - start_time)
64
65
66 print ("max_score")
67 print (max_score)
68 print ("max_k")
69 print (max_k)
70 print ("max_leaf_size")
71 print (max_leaf_size)
72 print ("max_p")
73 print (max_p)
74
75 print ("end of the script")
76 end_time = datetime.datetime.now()
77 print(end_time)
78 print("Total time:")
79 print (end_time - start_time)
```