

**FACULTAD
INFORMÁTICA MATEMÁTICA**

Aplicación web para la realización de experimentos de forma colaborativa en el framework JCLAL

**TRABAJO DE DIPLOMA PRESENTADO EN OPCIÓN
AL TÍTULO DE INGENIERO INFORMÁTICO**

Autor: Ramón Alejandro Valdés Ochoa
Tutores: M. Sc. Abel A. Fernández Higuera
M. Sc. Eduardo Pérez Perdomo

HOLGUÍN 2018



PENSAMIENTO

"Lo que nos hace verdaderamente grandes es el hecho de reconocer lo pequeños que somos".

Ramón Alejandro Valdés Ochoa.

DEDICATORIA

A mi abuela; la persona que sé que estará más orgullosa de mí en este momento, la que me impulsó a optar por recorrer el camino de superación y desarrollo que es el estudio de esta ciencia, en el que descubrí mi gran pasión. A ella especialmente está dedicado este trabajo.

A mis padres por estar siempre ahí para mí, entregándome todo sin pedir nada a cambio.

A mi esposa, por su compañía y dedicación en todo este tiempo.

A mi hermano Lisandro, por formar parte fundamental en mi desarrollo como ingeniero. Gracias por creer siempre en mí.

No existe forma de expresarles mi eterno amor y agradecimiento.

AGRADECIMIENTOS

Toda obra humana tiene muchas personas que la han alentado, este es el caso y aunque solo mencionaré a los más representativos, en mi agradecimiento hay lugar para todos.

A mi tutor Abel por introducirme en el maravilloso mundo de la web, por compartir sus conocimientos y experiencia y enseñarme a realizar los trabajos con profesionalidad y buen diseño.

A mi tutor Eduardo, la persona que me enseñó a programar con buenas prácticas. Gracias por ser tan exigente y por confiar en mí.

A mis amigos Leo y Elizabeth por su gran apoyo.

A Luis Ángel, muchas gracias por la confianza que siempre has depositado en mí.

A Yorlay, Franger, Rafa y Luis Miguel por sus consejos.

A Arles por mostrarme el valor de la física y la matemática para un ingeniero.

A mi familia, por el optimismo, el ánimo, la perseverancia y apoyo constante.

A mis amistades, por sus consejos, aientos y alegrías compartidas en todo este trayecto.

Al colectivo de profesores, por contribuir con sus conocimientos en mi formación profesional, por su exigencia y profesionalidad.

A los que de una forma u otra me han alentado en este camino.

A todos, GRACIAS

RESUMEN

El avance que supuso la creación de la primera red posibilitó el posterior nacimiento de la internet, que, en la actualidad, es el medio fundamental por el cual se mueve la información a través del mundo, lo que posibilita que cada vez sea más popular el conjunto de tecnologías y servicios basados en la web.

La Inteligencia Artificial se divide en varias ramas y una de ellas es el Aprendizaje Automático; que tiene como objetivo crear técnicas que le permitan a las computadoras aprender, generalizando patrones a partir de conocimiento adquirido anteriormente. El estudio de este se divide en varias áreas entre las que se encuentra el Aprendizaje Supervisado, No Supervisado, Semi-supervisado y el Aprendizaje Activo. Este ultimo mejora la precisión y el rendimiento general de los modelos predictivos.

JCLAL es un *framework* de Aprendizaje Automático que cuenta con muchos de los algoritmos de Aprendizaje Activo más populares que existen hoy en día. Entre sus principales características se encuentran su portabilidad, interoperabilidad, elegancia y una interfaz grafica que permite un mejor manejo del mismo. Aunque el framework en esta etapa es completamente funcional; la manera en la que se está moviendo el mundo en la adopción de este tipo de tecnologías hace que la evolución lógica en el su desarrollo sea dotarlo de la posibilidad de llegar a más investigadores alrededor del mundo y que estos puedan trabajar entre si de manera colaborativa aprovechando las bondades que brinda el maravilloso mundo de la web.

En la presente investigación se propone el desarrollo de una aplicación web para el uso del framework JCLAL en entornos colaborativos. Se analizan los elementos teóricos que sustentan el Aprendizaje Activo, el framework JCLAL y las tecnologías web actuales. Se diseña la estructura de la aplicación y se aplica al desarrollo de la misma. Se realizan pruebas de aceptación para comprobar la usabilidad de la solución propuesta.

ABSTRACT

The advance that led to the creation of the first network enabled the subsequent birth of the Internet, which, at present, is the fundamental means by which information moves through the world, which makes it possible for it to become increasingly popular. set of web-based technologies and services.

The Artificial Intelligence is divided into several branches and one of them is the Machine Learning; which aims to create techniques that allow computers to learn, generalizing patterns from previously acquired knowledge. The study of this is divided into several areas among which is Supervised, Unsupervised, Semi-supervised and Active Learning. The latter improves the accuracy and overall performance of predictive models.

JCLAL is a Machine Learning framework that has many of the most popular Active Learning algorithms that exist today. Among its main features are its portability, interoperability, elegance and a graphical interface that allows a better management of it. Although the framework in this stage is fully functional; The way in which the world is moving in the adoption of this type of technology makes, the logical evolution in its development is to give it the possibility of reaching more researchers around the world and that they can work together collaboratively, taking advantage of the benefits offered by the wonderful world of the web.

In this research its proposed the development of a web application for the use of the JCLAL framework in collaborative environments. Its been analyzed the theoretical elements that support Active Learning, the JCLAL framework and current web technologies. The structure of the application is designed and applied to the development of it. Acceptance tests are carried out to verify the usability of the proposed solution.

Tabla de contenido

Introducción.....	12
Capítulo 1: Fundamentos teóricos en el desarrollo de la aplicación web para el framework JCLAL.....	17
1.1 Aprendizaje Automático	17
1.1.1 Aprendizaje Supervisado	17
1.1.2 Aprendizaje No Supervisado	19
1.1.3 Aprendizaje Semi-supervisado	19
1.1.2 Aprendizaje Activo.....	19
1.1.2.1 Protocolos de aprendizaje activo	21
1.1.2.2 Estrategias de consulta	22
1.2 El Framework JCLAL.....	22
1.2.1 VisualJCLAL	24
1.3 Aplicación Web	25
1.3.1 Diseño de una aplicación web	26
1.4 Tecnologías y herramientas.....	27
1.4.1 Lenguajes de Programación	27
1.4.1.1 HTML.....	27
1.4.1.2 CSS	28
1.4.1.3 JavaScript.....	28
1.4.1.4 Java	29
1.4.1.5 XML.....	30
1.4.2 Entorno de Desarrollo Integrado.....	31
1.4.2.1 Visual Studio Code.....	31
1.4.2.2 Node.js	32
1.4.2.3 Express.js	33
1.4.2.4 jQuery.....	34
1.4.2.5 Bootstrap 4.....	35
1.4.2.6 Mongo DB	36
1.4.2.7 Git.....	37
1.4.3 Metodologías de desarrollo de Software.....	38
1.4.3.1 La metodología de desarrollo XP	39
1.5 Conclusiones parciales	40
Capítulo 2: Diseño e implementación de la aplicación web propuesta.....	41
2.1 Definición de requerimientos	41
2.1.1 Requerimientos funcionales	41

2.1.2 Requerimientos no funcionales	42
2.1.3 Personas relacionadas con la utilización de la aplicación web	43
2.1.4 Historias de Usuario.....	43
2.2 Planeación	48
2.2.1 Plan de iteraciones.....	48
2.2.2 Plan de entregas	49
2.3 Arquitectura del sistema.....	50
2.4 Implementación.....	51
2.4.1 Iteración 1.....	51
2.4.2 Iteración 2	53
2.4.3 Iteración 3	55
2.4.4 Iteración 4	56
2.4.5 Iteración 5	57
2.4.6 Iteración 6	60
2.5 Prueba	61
2.5.1 Pruebas de Aceptación	62
2.6 Conclusiones Parciales.....	66
Conclusiones.....	67
Recomendaciones.....	68
Referencias bibliográficas	69

Índice de Figuras

Figura 1: SL tradicional mono-etiqueta.....	18
Figura 2: Aprendizaje por multi-etiqueta.....	18
Figura 3: Aprendizaje por multi-instanciación multi-etiqueta	18
Figura 4: Aprendizaje por multi-instanciación	18
Figura 5: Ciclo de Aprendizaje Activo.....	20
Figura 6 VisualJCLAL. Vista de configuración de un experimento	25
Figura 7: Arquitectura de la aplicación web	50

Índice de Tablas

Tabla 1: Historia de Usuario No. 1 - Crear estructura de la aplicación web	44
Tabla 2: Historia de Usuario No. 2 - Configurar nuevo experimento.....	44
Tabla 3: Historia de Usuario No.3 - Crear una configuración dinámica de los experimentos.....	45
Tabla 4: Historia de Usuario No.4 - Gestionar usuarios.....	45
Tabla 5: Historia de Usuario No.5 - Guardar la configuración de un experimento.	46
Tabla 6: Historia de Usuario No.6 - Cargar la configuración de un experimento.	46
Tabla 7: Historia de Usuario No.7 - Ejecutar un experimento.....	47
Tabla 8: Historia de Usuario No.8 - Visualizar resultados de un experimento.....	47
Tabla 9: Historia de Usuario No.9 - Guardar historial de experimentación.....	48
Tabla 10: Distribución de las historias de usuario por iteración.	49
Tabla 11: Tarea 1: Crear prototipo no funcional de la aplicación web.....	51
Tabla 12: Tarea 2: Crear vista principal.....	52
Tabla 13: Tarea 3: Crear vista de configuración de experimentos.....	52
Tabla 14: Tarea 4: Crear vista de edición de código	52
Tabla 15: Tarea 5: Crear vista de login.....	53
Tabla 16: Tarea 6: Crear vista de usuario	53
Tabla 17: Tarea 7: Implementar la forma en que se genera el XML	54
Tabla 18: Tarea 18: Implementar configuración predeterminada de un experimento.	54
Tabla 19: Tarea 19: Implementar configuración dinámica de un experimento... ..	54
Tabla 20: Tarea 10: Tarea 10: Implementar formulario de registro de usuario	55
Tabla 21: Tarea 11: Implementar autentificación del login.....	55
Tabla 22: Tarea 12: Implementar sistema de grupos de usuarios.....	56
Tabla 23: Tarea 13: Implementar la forma en la que se guarda una configuración en la base de datos	56
Tabla 24: Tarea 14: Implementar la forma en la que se carga una configuración desde la base de datos	57
Tabla 25: Tarea 15: Implementar la manera en la que el sistema interactúa con el framework JCLAL.....	58
Tabla 26: Tarea 16: Implementar la manera de enviar el conjunto de datos al framework JCLAL.....	58
Tabla 27: Tarea 17: Ejecutar un experimento usando el framework JCLAL.....	58
Tabla 28: Tarea 18: Ejecutar un experimento en paralelo usando el framework JCLAL	59

Tabla 29: Tarea 19: Ejecutar un experimento de forma distribuida usando el framework JCLAL.....	59
Tabla 30: Tarea 20: Implementar la manera de procesar los datos de los resultados de un experimento	59
Tabla 31: Tarea 21: Implementar la forma en la que se muestran los resultados de un experimento de forma gráfica	60
Tabla 32: Tarea 22: Implementar la forma de guardar los resultados en la base de datos	60
Tabla 33: Tarea 23: Implementar la manera de que el usuario pueda gestionar el historial de experimentación.....	61
Tabla 34: Tarea 24: Implementar la manera en que se maneja la privacidad del historial de experimentación de cada usuario	61
Tabla 35: Caso de Prueba de Aceptación UH2-P1.....	62
Tabla 36: Caso de Prueba de Aceptación UH3-P2	63
Tabla 37: Caso de Prueba de Aceptación UH4-P3	63
Tabla 38: Caso de Prueba de Aceptación UH5-P4	64
Tabla 39: Caso de Prueba de Aceptación UH6-P5	64
Tabla 40: Caso de Prueba de Aceptación UH7-P6	65
Tabla 41: Caso de Prueba de Aceptación UH8-P7.....	65
Tabla 42: Caso de Prueba de Aceptación UH9-P8.....	66

Introducción

La Inteligencia Artificial (IA por sus siglas en español) es actualmente una de las tecnologías con más rápido nivel de desarrollo, su implementación está cambiando el mundo en casi todos los sectores. La capacidad de optimizar y automatizar decisiones en tiempo real está permitiendo una mejora radical en todos los ámbitos.

Aunque sus conceptos básicos fueron planteados desde mediados el siglo pasado por Alan Turing¹ no fue hasta finales del mismo que se produjeron avances significativos como el que propuso la empresa IBM con un algoritmo que corría en la supercomputadora DeepBlue que fue capaz de batir a Kasparov² en una partida de ajedrez. En el presente siglo otras importantes compañías como Apple, Google y Microsoft se han volcado de lleno en la investigación y desarrollo de sistemas que integren tecnologías de IA. A partir del 2011 se ha producido un avance vertiginoso en este campo; surgiendo una inmensa variedad de herramientas y aplicaciones.

EL estudio de la IA se divide en varias ramas dentro de las que se encuentra el **Aprendizaje Automático** (*Machine Learning*, ML, por sus siglas en inglés) en la que se incluye el desarrollo de algoritmos y técnicas que permitan al sistema aprender de forma automática basado en análisis de casos que generen experiencia previa. Entre sus aplicaciones se encuentran sistemas de clasificación de imágenes, sistemas de inferencia, modelos predictivos y probabilísticos, procesamiento de voz y reconocimiento de escritura, motores de búsqueda, análisis de mercado, sistemas de toma de decisiones en diagnósticos médicos y muchas otras. Para su funcionamiento es necesario el análisis de grandes volúmenes de datos para lo cual se utilizan distintos tipos de algoritmos que, en función a su funcionamiento se clasifican en su mayoría como **aprendizaje supervisado** (SL, por sus siglas en inglés, *Semi-Supervised Learning*), **aprendizaje no supervisado** y **semi-supervisado** (SSL, por sus siglas en inglés, *Supervised Learning*). En el aprendizaje supervisado la información que se utiliza está previamente clasificada y se conoce la categoría a la que pertenece cada ejemplo [1]. En el aprendizaje no supervisado no existe una clasificación previa de la información, la cual se agrupa automáticamente.

Para la clasificación de nuevos casos se precisa de un modelo predictivo lo suficientemente preciso para sustituir a un humano por lo que es necesario contar

¹ Considerado uno de los padres de la ciencia de la computación y precursor de la informática moderna

² Gary Kasparov es uno de los maestros ajedrecistas más prolíficos de la historia del deporte

con una amplia base de casos previamente etiquetados para la etapa de entrenamiento. Esta tarea se realiza por expertos humanos por lo que se convierte en un proceso costoso e ineficiente. Debido a lo poco práctico de este proceso surgen otras técnicas como el aprendizaje semi-supervisado y el aprendizaje activo³ (AL, por sus siglas en inglés, *Active Learning*).

Con el AL se intenta mejorar la precisión de un clasificador disminuyendo la cantidad de ejemplos previamente etiquetados que se necesitan, permitiéndole al algoritmo seleccionar los datos desde donde aprende y reduciendo el volumen necesario en el proceso de entrenamiento para lograr un mejor rendimiento con un menor costo de entrenamiento. Está motivado por muchos problemas modernos de ML donde los datos pueden ser abundantes, pero las etiquetas son escasas o costosas de obtener.[2]

Actualmente en el área del AL existen varios *frameworks* que apoyan el proceso de experimentación y desarrollo de nuevos algoritmos. Un framework en el área de las ciencias de la computación se define como un conjunto de técnicas y herramientas que permiten el desarrollo de algún producto, abstrayendo y facilitando el desarrollo de ciertas tareas según el dominio para el cual está construido el mismo[1].

Hasta la fecha, existen varias herramientas de software que ayudan al proceso de experimentación y al desarrollo de nuevos algoritmos en el área de ML, centrándose en el aprendizaje supervisado y no supervisado. Sin embargo, las herramientas de software para las áreas de AL y SL son escasas. Las pocas bibliotecas existentes para AL han sido diseñadas para resolver problemas específicos. No están diseñadas para un uso fácil, modificación, adaptación y extensión. Consecuentemente, los investigadores de AL tienen la carga de implementar trabajos relacionados, perdiendo un tiempo considerable en la experimentación y el desarrollo de nuevos métodos de AL. La situación antedicha motivó el desarrollo del framework JCLAL⁴ (*Java Class For Active Learning* por sus siglas en inglés).

JCLAL es un software de código abierto para que los investigadores y los usuarios finales desarrollen métodos de AL. JCLAL incluye las estrategias de consulta más relevantes que se han propuesto en paradigmas de aprendizaje de etiqueta única y multi-etiqueta. Proporciona las interfaces, las clases y los métodos necesarios para desarrollar cualquier método de AL.[3]

³ Se le conoce también como aprendizaje por consulta o diseño experimental óptimo en la literatura de estadística

⁴ <http://sourceforge.net/projects/jclal/files/>

La forma de realizar los experimentos en JCLAL es mediante la configuración de un archivo en formato XML donde se incluyen las etiquetas necesarias para la ejecución de los experimentos. Estas etiquetas hacen referencia a los principales elementos del flujo de AL, que son configurables mediante los paquetes de clases de JCLAL. Para ejecutar un experimento se deben usar distintos comandos desde la consola, que permitan al framework conocer los parámetros necesarios para desarrollar este proceso como son la dirección donde se encuentra el archivo de configuración, los métodos que se utilizaran en la experimentación o si se desea ejecutar el experimento en paralelo.[4]

En principio la experimentación con el framework era totalmente funcional; la interacción poco asistida en la configuración, la ejecución de los experimentos y comparación de los resultados obtenidos, provocaba aumento en el tiempo de trabajo, los investigadores debían dominar los paquetes de clases del framework, así como todas las etiquetas de XML⁵ necesarias para realizar los experimentos, además la posibilidad de introducir errores de escritura en la configuración del XML y tener problemas en encontrar su origen podían dificultar mucho su uso. Era poco probable que usuarios con poca habilidad en el uso del framework o con conocimientos básicos de AL, fueran capaces de realizar los experimentos de forma rápida y sencilla.

A partir de estas deficiencias surge VisualJCLAL⁶; un entorno de usuario para el framework JCLAL que facilitaba mucho la configuración de nuevos experimentos. VisualJCLAL cuenta con un diseño de pestañas que permite configurar paso a paso un experimento de AL y mostrar los resultados posteriormente.

Aunque VisualJCLAL agilizó mucho el proceso de experimentación con JCLAL; algunos aspectos mejorables como la necesidad de tener instalada la maquina virtual de Java en el terminal donde se valla a ejecutar, la configuración algo confusa para los usuarios poco experimentados, así como la imposibilidad de realizar una gestión de usuario que brindaría posibilidades como el almacenamiento de historiales de experimentación y el trabajo colaborativo en general entre varios investigadores en un entorno de red o localmente; conllevarían a la siguiente evolución de esta potente herramienta, y, de esta forma permita su acercamiento a más usuarios y facilite aún más el proceso de aprendizaje y familiarización con JCLAL y el área de AL en general.

⁵ XML (eXtensible Markup Language), "Lenguaje de Marcado Extensible", es un meta-lenguaje que permite definir lenguajes de marcas utilizado para almacenar datos en forma legible.

⁶ https://github.com/eperezp1990/jclal-develop/tree/visual_jclal

A partir de esta situación problemática anterior se puede identificar el siguiente **problema científico**: ¿Cómo facilitar la realización de experimentos usando el *framework* JCLAL en entornos colaborativos?

El **objeto de estudio** en el que se encuentra almacenado del problema es la experimentación usando el *framework* JCLAL.

Para darle solución a la deficiencia detectada en el problema científico se propone como **objetivo general**: desarrollar una aplicación web para el uso del *framework* JCLAL que permita la realización de experimentos de forma individual o colaborativa en un ambiente de desarrollo local o en un entorno de red.

El **campo de acción** delimitado por el objetivo general es: desarrollo de experimentos el usando *framework* JCLAL de forma colaborativa en una aplicación web.

Para guiar la investigación se trazaron las siguientes **preguntas científicas**:

¿Cuáles son los fundamentos teóricos de las técnicas de *Active Learning*?

¿Cuál es el uso actual de las técnicas y métodos de *Active Learning* en el *framework* JCLAL?

¿Cómo diseñar un entorno web que facilite la realización de experimentos en el *framework* JCLAL?

¿Cómo valorar la facilidad de uso del entorno web propuesto para el *framework* JCLAL?

En función de darle cumplimiento al objetivo de la investigación y de darle respuesta a las anteriores preguntas científicas se trazaron las siguientes **tareas científicas**:

- Analizar los fundamentos teóricos que rigen las técnicas de *Active Learning*.
- Analizar el uso actual de las técnicas de *Active Learning* en el *framework* JCLAL.

- Implementar una aplicación web para el uso del *framework* JCLAL, que facilite la realización de experimentos usando técnicas de *Active Learning en entornos colaborativos*.
- Valorar la usabilidad de la aplicación web propuesta.

Para dar solución a las tareas planteadas, se utilizó una combinación de métodos de trabajo científico, entre los que destacan los siguientes:

Métodos Teóricos: El método de análisis y síntesis se utilizó para descomponer el problema de investigación mentalmente en sub-problemas y profundizar en su estudio, para luego sintetizarlos en una solución que los integrara. El método de modelación permitió realizar el diseño de la aplicación web del *framework* JCLAL y de esta manera comprender mejor los elementos que se deben implementar. El método sistémico fue utilizado para crear la estructura de la aplicación web, en el análisis y determinación de las relaciones y dependencias. El histórico-lógico, para el estudio de manera cronológica de investigaciones previas y para usarse como elemento de referencia.

Métodos Empíricos: El método de revisión de la documentación permitió realizar una revisión bibliográfica profunda para plasmar y referenciar el conocimiento adquirido relacionado con el objeto de estudio y la fundamentación de la solución propuesta. El método de observación y el método de entrevista permitieron apreciar y entender la forma actual en que se realizan los experimentos usando el *framework* JCLAL.

El documento está estructurado en introducción, dos capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. En el Capítulo 1 se exponen las bases teóricas que dan fundamento a la investigación, y a continuación se describen las tecnologías y herramientas que se utilizan para el desarrollo de la aplicación web propuesta. El Capítulo 2 abarca lo referente a las acciones realizadas en las diferentes etapas que propone la metodología de desarrollo empleada, además se realiza una valoración de la calidad del software basado en la usabilidad del mismo mediante encuestas a los usuarios. Por último, se exponen las principales conclusiones a las que se arribaron con el desarrollo de esta investigación.

Capítulo 1: Fundamentos teóricos en el desarrollo de la aplicación web para el framework JCLAL

En este capítulo se plantean los conceptos que enmarcan el objeto de estudio y la solución propuesta, tales como las técnicas y métodos de AL. Además de su uso y aplicación en el framework JCLAL, abundando también en lo referente al diseño e implementación de aplicaciones web. Además, se describen las tecnologías y herramientas utilizadas en el desarrollo de la solución propuesta y se fundamenta la metodología de desarrollo escogida.

1.1 Aprendizaje Automático

Según el Diccionario General de la Lengua Española, aprender es adquirir el conocimiento de algo por medio del estudio, el ejercicio o la experiencia. Más centrado en el ámbito de las ciencias de la computación, el premio novel en economía Herbert Simon⁷ diría que el aprendizaje es cualquier proceso por el cual un sistema mejora el rendimiento basado en la experiencia". ML es una rama de la AI que tiene como objetivo desarrollar técnicas y algoritmos que otorguen a las computadoras la capacidad de aprender. Específicamente, intenta crear programas capacitados para generalizar comportamientos a partir de datos no estructurados, proporcionados en forma de ejemplos. Es un proceso inductivo⁸ del conocimiento[5]. Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de rendimiento P, si su desempeño en tareas T, medida por P, mejora con la experiencia E [6].

Esta área del AI tiene amplias aplicaciones ya que es utilizada en la clasificación de textos y multimedia, el procesamiento de lenguaje natural, descubrimiento de conocimiento valioso en grandes bases de datos comerciales, clasificación de imágenes, solicitudes de préstamo, transacciones bancarias y similares [6]. En la literatura se definen dos grandes grupos derivados de ML, el SL y el UL. En el primero se encuentra, entre otras tareas la clasificación y la regresión, y en el segundo tipo la de agrupamiento, también conocido como *clustering* [1].

1.1.1 Aprendizaje Supervisado

En el SL la información se encuentra previamente etiquetada y es requerido un gran número de datos para poder entrenar un clasificador lo suficientemente

⁷ Herbert Alexander Simon, Premio Turing 1975 y Premio Novel de Economía 1978

⁸ El proceso o aprendizaje inductivo, parte de casos particulares (ejemplos) y obtiene casos generales (reglas o modelos) que generalizan o abstraen la evidencia.

exacto [1]. Los algoritmos trabajan intentando encontrar una función que, dadas las variables de entrada, les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un “histórico” de datos y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida. En la literatura se pueden encontrar varios tipos de clasificación:

- De una etiqueta: Es la clasificación tradicional en la que la instancia tiene asociada una clase.
- Multi-etiqueta: La instancia puede pertenecer simultáneamente a distintas clases [7]
- Multi-instancias: Cuando el objeto es muy complejo, este puede ser representado por varias instancias y todas ellas pertenecer a la misma categoría. Ejemplo de ello es al dividir un documento en distintas secciones, por lo que cada sección estaría descrita por una instancia [8]
- Multi-instancias Multi-etiqueta: Es una mezcla de los dos tipos de aprendizajes anteriores. Este considera las ambigüedades tanto en los objetos como en sus categorías

Las siguientes ilustran lo planteado anteriormente.



Figura 1: SL tradicional mono-etiqueta

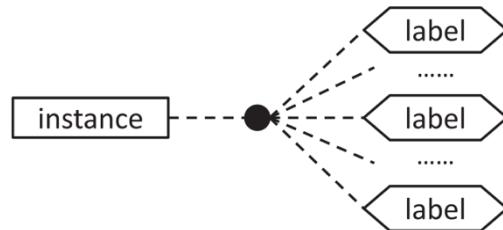


Figura 2: Aprendizaje por multi-etiqueta

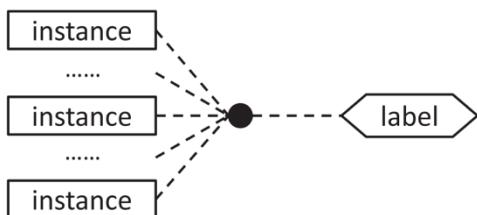


Figura 4: Aprendizaje por multi-instancias

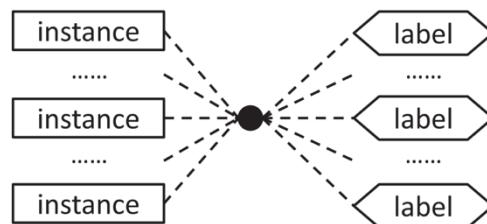


Figura 3: Aprendizaje por multi-instancias multi-etiqueta

1.1.2 Aprendizaje No Supervisado

En el aprendizaje no supervisado no existe una clasificación previa de ejemplos, por lo que estos son agrupados automáticamente. Por esta razón el agrupamiento es una de sus tareas. La tarea de agrupamiento tiene como objetivo formar una colección de grupos o subconjuntos, que cumplan con las propiedades de la homogeneidad dentro de los grupos y la heterogeneidad entre los grupos. En la primera, los ejemplos que pertenecen al mismo grupo deben ser tan similares como se pueda; en la segunda, los ejemplos que corresponden a grupos diferentes deben ser tan disímiles como se pueda. En otras palabras, la tarea de agrupamiento trata de seguir el principio de maximizar la similitud de los ejemplos en cada grupo y minimizar la similitud entre los grupos [1]

1.1.3 Aprendizaje Semi-supervisado

El propósito de los investigadores en el ámbito del aprendizaje supervisado, en específico en la tarea de clasificación, es lograr entrenar un clasificador capaz de sustituir a un experto humano. Para ello, es de vital importancia disponer de un número grande de ejemplos etiquetados para la etapa de entrenamiento, lo cual permite construir un modelo predictivo lo suficientemente preciso. En el mundo real existen disímiles escenarios donde se disponen de un número reducido de ejemplos etiquetados en conjunto con un mayor número de ejemplos sin clasificar. Ante este tipo de problemas las técnicas clásicas del aprendizaje supervisado resultan ineficientes.[1]

Una solución a esta problemática fue la fusión de los algoritmos del aprendizaje supervisado con los del no supervisado, el cual se denominó en la literatura aprendizaje semi-supervisado (SSL, por sus siglas en inglés, Semi-Supervised Learning). SSL es un paradigma del aprendizaje automático, en el cual el modelo es construido con un número pequeño de ejemplos etiquetados y un número grande de ejemplos no etiquetados [2]

1.1.2 Aprendizaje Activo

En el contexto del aprendizaje automático se plantea la necesidad de encontrar modos de entrenar un clasificador cuando el conjunto de entrenamiento es combinado con un número pequeño de ejemplos etiquetados y un gran número de no etiquetados. Como se ha mencionado anteriormente los métodos tradicionales de aprendizaje supervisado o no supervisado no satisfacen resolver tales problemas [1].

El aprendizaje activo en la tarea de clasificación puede mejorar el clasificador con menos ejemplos etiquetados de entrenamiento, si se permite elegir activamente los ejemplos de los cuales él aprende [2]. Es decir, el algoritmo de AL funciona tomando como entrada un pequeño conjunto de ejemplos etiquetados y uno más grande de ejemplos no etiquetados de los cuales selecciona de forma activa ejemplos a etiquetar.



Figura 5: Ciclo de Aprendizaje Activo

La aplicación del aprendizaje activo ha sido enfocada en amplia variedad de tareas entre las que se destacan:

- Clasificación y filtrado de imágenes, archivos de audio y video [3]
- Clasificación de texto [9]
- Procesamiento del lenguaje natural
 - Filtrado de información adaptable [10]
 - Extracción de información [11]
 - Reconocimiento de entidades [12]
 - Etiquetamiento de partes de la oración [13], [14]

- Análisis sintáctico [15], [16]
- Desambiguación del significado de la palabra [17]
- Comprensión del lenguaje hablado [18]
- Transliteración automática [19]
- Segmentación de secuencias [20]

El aprendizaje activo posee diferentes técnicas en el proceso de aprendizaje. De manera general estas técnicas funcionan dividiendo el conjunto de datos en uno de entrenamiento y otro de prueba. Este proceso se repite utilizando el conocimiento recién ganado para escoger cuidadosamente los próximos ejemplos a seleccionar. [4]

1.1.2.1 Protocolos de aprendizaje activo

Existen varios enfoques y protocolos en la literatura que describen la aplicación de estas técnicas. Algunas de las cuales serán resumidas a continuación:

Membership Query Synthesis (Síntesis de Consulta de Membresía): se consulta aleatoriamente cada ejemplo no etiquetado en el espacio de entrada, para luego ser etiquetado. [2], [21]

Stream-based Active Learning (Aprendizaje Activo basado en flujo): Este evalúa un ejemplo a la vez, para luego decidir si el ejemplo consultado es etiquetado o ignorado [22]

Pool-based Active Learning (Aprendizaje Activo basado en conjunto o piscina): dado un conjunto de ejemplos no etiquetados, los ordena de acuerdo a su informatividad y luego consulta la etiqueta para aquellos ejemplos que sean más informativos [2], [23]

El escenario síntesis de consulta de membresía es usualmente eficiente, pero el etiquetamiento aleatorio de ejemplos puede ser poco práctico para problemas de dominio real si el oráculo⁹ es un anotador humano, esta limitante dio lugar al uso de los otros dos escenarios.

La diferencia principal entre el aprendizaje activo basado en flujo y el basado en conjunto es que el primero realiza la exploración por los ejemplos secuencialmente y toma decisiones de consulta de manera individual, mientras

⁹ En aprendizaje activo un oráculo es aquel que se encarga de etiquetar los ejemplos no etiquetados, este puede ser un humano o simulado.

que el otro evalúa y ordena el fondo completo de ejemplos no etiquetados antes de seleccionar las mejores consultas. [2]

El protocolo basado en fondo ha sido mucho más aplicado en tareas reales como en clasificación y recuperación de imágenes [24], [25], clasificación y recuperación de videos [26], clasificación de texto [25], [26], extracción de información [27], [28], reconocimiento del habla [18], [29] y en el diagnóstico de cáncer [30].

El protocolo basado en flujo es más apropiado cuando la memoria o el poder de procesamiento pueden ser limitados, como con los dispositivos móviles e incrustados. Es aplicado en tareas como el reconocimiento de palabras [13], programación de sensores [2] y en el área de recuperación de la información [2].

1.1.2.2 Estrategias de consulta

Como se explicó en el epígrafe anterior una de las partes importantes dentro de las técnicas de aprendizaje activo es el método de consulta de la etiqueta de los ejemplos a etiquetar. De ahí, que en todos los protocolos anteriormente explicados se emplee el término consulta como proceso de evaluación de información de los ejemplos no etiquetados.

En la bibliografía existen diferentes criterios para formular las estrategias de consulta, tal es el caso de:

- Muestra de incertidumbre [2]
- Consulta por comité [2], [31], [32]
- Reducción esperada del error [2]
- Reducción de la varianza [2], [26], [33]
- Densidad y diversidad [2], [27]

1.2 El Framework JCLAL

JCLAL es un framework de código abierto bajo la licencia GNU/GPL (General Public License, en inglés) para el desarrollo de técnicas de AL. Fue desarrollado en la Universidad de Holguín y está inspirado en la arquitectura del framework JCLEC (Java Class Library for Evolutional Computing, en inglés) [34] para la computación evolutiva. Posee una arquitectura que sigue los principios de la programación orientada a objetos, donde es común y fácil reutilizar código.

JCLAL implementa dos de los escenarios de AL, el Muestreo Selectivo basado en Flujo y el basado en fondo. Por su estructura, es muy simple definir un conjunto

de *listeners* para responder según los eventos desencadenados por un algoritmo. Ejemplo de ello son los que realizan los reportes sobre las métricas extraídas en resultados de los métodos de AL y otro para la visualización gráfica de estos resultados.

Para la realización de pruebas a los algoritmos cuenta con los métodos de evaluación *Hold-Out*, *K-Fold Cross Validation*, *Five-Per-Two Cross Validation (5x2 CV)* y *Leave-One-Out*. [3]

Brinda reconocidas estrategias de consulta para problemas de una sola instancia y multi-etiqueta (para el paradigma multi-etiqueta está apoyado en la biblioteca de clases MULAN [35]). Para las estrategias de consulta de una sola instancia tenemos a *Entropy Sampling*, *Least Confident* y *Margin Sampling* pertenecientes a *Uncertainty Sampling* y dentro de las *Query By Committee a Vote Entropy* y *Kullback Leibler Divergence*. Posee las dos variantes de *Expected Error Reduction*, *Expected 0/1-loss* y *Expected Log-Loss*. Presenta una estrategia que pertenece a la rama de *Variance Reduction* y el framework de *Information Density*. Dentro de las estrategias multi-etiqueta están *Binary Minimum*, *Max Loss*, *Mean Max Loss*, *Maximal Loss Reduction* con *Maximal Confidence*, *Confidence-Minimum-NonWeighted* y *Confidence-Average-NonWeighted*. Todas ellas pueden ser usadas con cualquier clasificador base presentes en Weka y MULAN.

Dentro de las características principales podemos mencionar:

- **Genérico:** Permite incluir nuevos métodos de AL a través de una estructura de clases flexible, así como contar con la posibilidad de adaptar, modificar o extender el framework en función de las necesidades del desarrollador.
- **Portable:** el framework fue creado con código en el lenguaje de programación Java, asegurando su portabilidad a través de todas las plataformas que implementen un Máquina Virtual de Java.
- **Elegante:** el framework posee una estructura de clases coherente que sigue los buenos principios de la programación genérica y orientada a objetos.
- **Interoperable:** el uso de XML brinda una base común para las herramientas de desarrollo y para vincularlo con otros sistemas mediante el uso de wrappers que lo vuelven a la vez flexible pues permite la integración con cualquier otro framework.
- **Código abierto:** el código es libre y lanzado bajo la licencia GNU/GPL (*General Public License*), por tanto, puede ser distribuido y modificado sin

ningún cargo. Se puede encontrar en SourceForge¹⁰, GitHub¹¹, OSSRH repositorio proporcionado por Sonatype, y Maven Central Repository.

- **Distribuido:** mediante el uso del framework Spark permite ejecutar los experimentos de JCLAL en clústeres distribuidos en la red incrementando la velocidad de experimentación.

Igualmente presenta un conjunto de utilidades como algoritmos para la generación de números aleatorios, algoritmos de ordenamiento, métodos de muestreo y métodos para hallar el Área Bajo la Curva de Aprendizaje. Conjuntamente presenta clases para la configuración gráfica de los archivos de configuración XML y un plugin para la integración con el explorador de WEKA [3].

A partir de la versión 2.0 posee una arquitectura modular lo cual permite un mejor desacoplamiento del código y manejo de dependencias. El aprendizaje sobre conjunto de datos masivos fue añadido en el módulo jclal-spark, utilizando un procesamiento en memoria a través del conocido framework Apache Spark. Fueron incluidos una serie de test estadísticos no paramétricos como apoyo al proceso de experimentación. Por último, todas las funcionalidades anteriormente mencionadas están presentes en la nueva interfaz visual (jclal-gui).

1.2.1 VisualJCLAL

Actualmente JCLAL cuenta con una GUI (por sus siglas en inglés, *Graphical User Interface*, Interfaz Grafica de Usuario), de la cual podemos observar un ejemplo en la Figura 5, que fue construida sobre una arquitectura basada en componentes, donde, a través de plugin se pueden añadir nuevas funcionalidades al framework. Aunque JCLAL era totalmente funcional, problemas como: la interacción poco asistida en la configuración, ejecución de experimentos y comparación de los resultados obtenidos podía provocar que el tiempo de trabajo aumentara. Los investigadores debían dominar los paquetes de clases del framework, así como todas las etiquetas de XML necesarias para realizar los experimentos. Los usuarios con poca habilidad en el uso del framework o con conocimientos básicos de AL, era poco probable que fueran capaces de realizar los experimentos de forma rápida y sencilla o que se podían introducir errores de escritura en la configuración del XML y tener problemas en encontrar su origen dieron paso al desarrollo de esta.

¹⁰ <https://sourceforge.net/projects/jclal/files/>

¹¹ <https://github.com/ogreypesp/JCLAL>

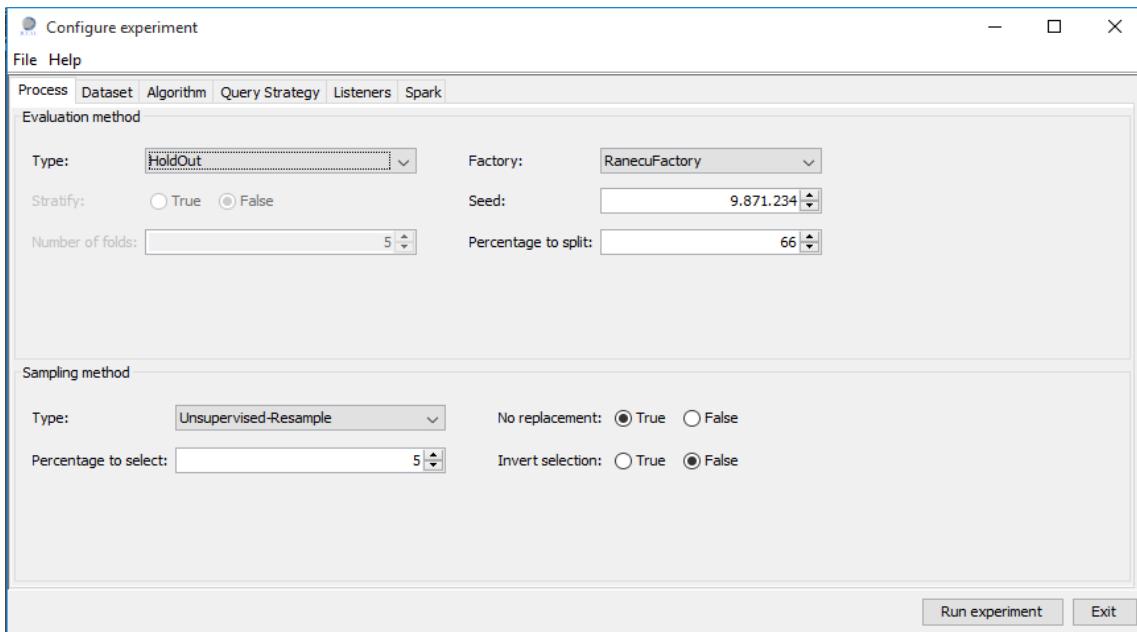


Figura 6 VisualJCLAL. Vista de configuración de un experimento

Aunque la implementación de esta GUI supuso un gran avance en el proceso de evolución del framework, todavía existían algunas deficiencias como que la maquina virtual de Java debe estar instalada en el terminal donde se fuera a usar, la configuración de los experimentos re realizaba de manera confusa para los investigadores con poca experiencia, no era almacenado un historial de experimentación con el cual comparar resultados, no permitía una gestión de usuarios ni la experimentación de forma colaborativa entre varios investigadores.

1.3 Aplicación Web

Una aplicación web es una aplicación o herramienta informática accesible desde cualquier navegador, bien sea a través de internet o bien a través de una red local. A través de el navegador se puede acceder a todas las funcionalidades con las que cuente dicha aplicación. [36]

Entre sus principales ventajas podemos encontrar:

- **Ahorro de tiempo:** se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- **Compatibilidad:** basta tener un navegador actualizado para poder utilizarla.
- **Espacio:** no ocupa espacio en el disco duro local.
- **Actualizaciones inmediatas:** como el software es gestionado por el propio desarrollador, el usuario estaría usando siempre la última versión que se haya lanzado.

- **Consumo de recursos bajo:** dado que toda (o gran parte) de la aplicación no se encuentra en el terminal, muchas de las tareas que realiza el software no consumen recursos de este porque se realizan desde el servidor que este ejecutando la aplicación.
- **Multiplataforma:** se pueden usar desde cualquier sistema operativo porque solamente es necesario tener un navegador.
- **Portables:** es independiente del terminal donde se utilice porque se accede a través de una página.
- **Virus/malware:** son menos proclives a ser afectados por los virus o programas maliciosos de cualquier índole instalados en el equipo, aunque no sean inmunes a ellos.
- **Disponibilidad:** suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- **Colaboración:** gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios.
- **Aumento de funcionalidades:** los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web enriquecidas.

1.3.1 Diseño de una aplicación web

Una aplicación Web típica incluye un servidor Web, un servidor de aplicación y un servidor de base de datos. Cada uno de ellos escucha a un puerto TCP/IP específico para comprobar si entran mensajes contenido solicitudes para gestionar. Estas escuchas se suelen llamar daemons. Son hilos de ejecución que esperan que aparezcan mensajes TCP/IP por un puerto específico. Los puertos proporcionan un método sencillo en el que cada empresa puede limitar el acceso a sus recursos en red. Los servidores de aplicación escuchan un puerto que normalmente es privado con respecto al mundo real. El servidor Web está configurado para cumplimentar solicitudes específicas a dicho puerto en particular. Los servidores de base de datos operan de un modo similar. El servidor de base de datos puede estar escuchando en el puerto 3306 esperando solicitudes. El servidor de aplicación entonces establecería una conexión de base de datos con dicho puerto.

1.4 Tecnologías y herramientas

A continuación, serán descritas las herramientas y tecnologías que se analizaron y seleccionaron para el desarrollo de la solución propuesta de acuerdo con las características que debe poseer la aplicación web.

1.4.1 Lenguajes de Programación

Para el desarrollo de la aplicación web propuesta serán usados diferentes lenguajes de programación descritos a continuación.

1.4.1.1 HTML

HTML, (sigla en inglés de *HyperText Markup Language*, lenguaje de marcas de hipertexto), es un lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del *World Wide Web Consortium (W3C)*¹², organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la *World Wide Web (WWW)*. Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado. [37]

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado. [37]

Fue escogido este lenguaje por ser un estándar en el área de desarrollo web.

¹² <https://www.w3.org/html/>

1.4.1.2 CSS

CSS (siglas en inglés de *Cascading Style Sheets*, Hojas de estilo en cascada), es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS). [38]

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o *layouts*, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento. [37]

Fue escogido este lenguaje por ser un estándar en el área de desarrollo web y su integración con otras tecnologías utilizadas.

1.4.1.3 JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. [39] Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (*Client-Side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (*Server-Side JavaScript* o SSJS) como veremos más adelante. Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente *widgets*¹³) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos

¹³ En informática, un widget es una pequeña aplicación o programa ejecutado por un Widget Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

soportan por lo menos ECMAScript 3. JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (DOM) [40].

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML [40].

Fue escogido este lenguaje por su flexibilidad además de la amplia gama de tecnologías compatibles y el masivo soporte en línea que posee.

1.4.1.4 Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web. [41]

El lenguaje de programación Java fue originalmente desarrollado por James Gosling, de Sun Microsystems, y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a bytecode (clase Java), que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. [42]

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar el paradigma de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.

- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++

Entre los factores que influyeron en la elección de este lenguaje se encuentra:

- Compatibilidad con el framework JCLAL, el cual fue desarrollado en Java.
- La documentación y código fuente disponible hacen a este lenguaje mucho más accesible para los programadores.

1.4.1.5 XML

Siglas en inglés de *eXtensible Markup Language*, traducido como "Lenguaje de Marcado Extensible" o "Lenguaje de Marcas Extensible", es un meta-lenguaje que permite definir lenguajes de marcas desarrollado por el W3C utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. [43]

XML no nació únicamente para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande, con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

El uso de esta tecnología en la aplicación web propuesta está determinado por la capacidad del *framework* JCLAL de contener un conjunto de clases que proveen funcionalidades para visualizar los resultados e interfaces de usuarios para configurar experimentos en XML, así como las clases necesarias para ejecutar los experimentos mediante ficheros XML.

1.4.2 Entorno de Desarrollo Integrado

Para el desarrollo de la aplicación web se utilizó un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés, *Integrated Development Environment*). Un IDE es un programa informático compuesto por un conjunto de herramientas de programación, orientado a uno o varios lenguajes de programación para facilitar el desarrollo de cualquier aplicación a los programadores.

1.4.2.1 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft de potente pero ligero que está disponible para Windows, MacOS y Linux. Viene con soporte integrado para JavaScript, TypeScript, Node.js y tiene un rico ecosistema de extensiones para otros idiomas como C++, C#, Java, Python, PHP, Go y Runtimes como .NET y Unity. Se basa en Electron, un framework que se utiliza para implementar aplicaciones Node.js para el escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor "Monaco" utilizado en *Visual Studio Team Services* (anteriormente llamado *Visual Studio Online*)

Cuenta con potentes herramientas de desarrollo como el auto completamiento y depuración de código IntelliSense. VS Code también se integra con herramientas de creación de scripts para realizar tareas comunes que agilizan los flujos de trabajo. Es compatible con Git, por lo que se puede trabajar con el control de código fuente sin abandonar el editor, incluida la visualización de los cambios pendientes. También incluye herramientas excelentes para tecnologías web como JS / React, HTML, CSS, SCSS, y JSON por lo que lo hace de esta herramienta la ideal para la realización de la aplicación web planteada [44].

Entre las razones por las que se escogió esta tecnología para el desarrollo de la aplicación web propuesta se encuentran:

- La compatibilidad con HTML, CSS, XML, JavaScript, Node.js y Java que son otras tecnologías que serán usadas en el desarrollo de la aplicación web.
- Las posibilidades que brinda al ser completamente compatible con Git.
- El excelente rendimiento y estabilidad que brinda esta herramienta.
- Experiencia del desarrollador en el uso de esta herramienta.

1.4.2.2 Node.js

Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node está diseñado para construir aplicaciones en red escalables. Por cada conexión el callback será ejecutado, sin embargo si no hay trabajo que hacer Node estará durmiendo. Esto contrasta con el modelo de concurrencia más común hoy en día, donde se usan hilos del Sistema Operativo.

Las operaciones de redes basadas en hilos son relativamente ineficientes y son muy difíciles de usar. Además, los usuarios de Node están libres de preocupaciones sobre el bloqueo del proceso, ya que no existe. Casi ninguna función en Node realiza I/O directamente, así que el proceso nunca se bloquea. Debido a que no hay bloqueo es muy razonable desarrollar sistemas escalables en Node. [45]

Tiene un diseño similar y está influenciado por sistemas como *Event Machine* de Ruby o *Twisted* de Python. Node lleva el modelo de eventos un poco más allá, este presenta un bucle de eventos como un entorno en vez de una librería. En otros sistemas siempre existe una llamada que bloquea para iniciar el bucle de eventos. El comportamiento es típicamente definido a través de callbacks al inicio del *script* y al final se inicia el servidor mediante una llamada de bloqueo como `EventMachine::run()`. En Node no existe esta llamada. Node simplemente ingresa el bucle de eventos después de ejecutar el script de entrada. Node sale del bucle de eventos cuando no hay más callbacks que ejecutar. Se comporta de una forma similar a JavaScript en el navegador el bucle de eventos está oculto al usuario.

HTTP es ciudadano de primera clase en Node, diseñado con operaciones de *streaming* y baja latencia en mente. Esto hace a Node candidato para ser la base de una librería o un framework web. [45]

Solo porque Node esté diseñado sin hilos, no significa que no se pueda aprovechar los múltiples cores del sistema. Procesos hijos pueden ser lanzados usando la API `child_process.fork()`, la cual está diseñada para comunicarse fácilmente con el proceso principal. Construida sobre la misma interfaz está el módulo `cluster`, el cual permite compartir sockets entre procesos para activar el balanceo de cargas en sus múltiples cores. [45]

El siguiente conjunto de características justificó la elección de esta tecnología para el desarrollo de la solución propuesta:

- Node ha sido diseñado para optimizar el rendimiento y la escalabilidad en aplicaciones web y es un muy buen complemento para muchos problemas comunes de desarrollo web (ej, aplicaciones web en tiempo real).
- El código está escrito en JavaScript, lo que significa que se pierde menos tiempo ocupándose de las conmutaciones de contexto entre lenguajes cuando se está escribiendo tanto el código del explorador web como del servidor.
- El gestor de paquetes de Node (NPM del inglés: *Node Packet Manager*) proporciona acceso a cientos o miles de paquetes reutilizables.

1.4.2.3 Express.js

Express.js es un framework para Node.js que sirve para ayudarnos a crear aplicaciones web en menos tiempo ya que proporciona funcionalidades como el enrutamiento, opciones para gestionar sesiones y cookies entre otras. Está basado en Connect, que a su vez es un framework basado en HTTP para Node.js. Podemos decir que Connect tiene todas las opciones del módulo http que viene por defecto con Node y le suma funcionalidades. A su vez, Express hace lo mismo con Connect, con lo que tenemos un *framework* ligero, rápido y muy útil.

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros *frameworks* populares web de Node. A pesar de que Express es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. [46]

En sitios web o aplicaciones web dinámicas, que accedan a bases de datos, el servidor espera a recibir peticiones HTTP del navegador (o cliente). Cuando se recibe una petición, la aplicación determina cuál es la acción adecuada correspondiente, de acuerdo a la estructura de la URL y a la información (opcional) indicada en la petición con los métodos POST o GET. Dependiendo de la acción a realizar, puede que se necesite leer o escribir en la base de datos, o realizar otras acciones necesarias para atender la petición correctamente. La aplicación ha de responder al navegador, normalmente, creando una página HTML dinámicamente para él, en la que se muestre la información pedida, usualmente dentro de un elemento específico para este fin, en una plantilla HTML.

Express posee métodos para especificar que función ha de ser llamada dependiendo del verbo HTTP usado en la petición (GET, POST, SET, etc.) y la

estructura de la URL ("ruta"). También tiene los métodos para especificar que plantilla ("view") o gestor de visualización utilizar, donde están guardadas las plantillas de HTML que han de usarse y como generar la visualización adecuada para cada caso. El middleware de Express, puede usarse también para añadir funcionalidades para la gestión de cookies, sesiones y usuarios, mediante el uso de parámetros, en los métodos POST/GET. Puede utilizarse además cualquier sistema de trabajo con bases de datos, que sea soportado por Node (Express no especifica ningún método preferido para trabajar con bases de datos). [46]

Fue escogido para el trabajo en la aplicación web propuesta ya que proporciona mecanismos para:

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

1.4.2.4 jQuery

jQuery es una biblioteca de JavaScript rápida, pequeña y con muchas funciones cuyo objetivo es facilitar la programación scripting al ofrecer una serie de funciones y métodos con los cuales los usuarios pueden producir aplicaciones web o sitios web más rápido y fácil. Hace que cosas como la manipulación de documentos HTML, el manejo de eventos, la animación y AJAX¹⁴ sean mucho más simples con una API fácil de usar que funciona en una gran cantidad de navegadores. [47]

jQuery proporciona una capa de abstracción de propósito general para los scripts web comunes, y por lo tanto es útil en casi todas las situaciones de scripting. Su naturaleza es extensible ya que los complementos se desarrollan constantemente para agregar nuevas habilidades. [48]

¹⁴ AJAX (Asíncronos JavaScript And XML) es una técnica de desarrollo web para crear aplicaciones interactivas.

Es la biblioteca más empleada de JavaScript. JQuery fue creado por John Resig¹⁵ y lanzado oficialmente en el año 2006.

Se escogió por contar entre sus principales ventajas con:

- Su librería permite actualizaciones constantes y rápidas.
- Posee un código abierto y compatible con diferentes navegadores.
- Es fácil de usar, por lo que permite ahorrar tiempo y esfuerzo.
- Su desempeño se integra muy bien con AJAX.
- Permite realizar animaciones, efectos y personalizaciones.

1.4.2.5 Bootstrap 4

Es un conjunto de herramientas de código abierto para desarrollar con HTML, CSS y JS y crear proyectos adaptables y móviles en la web. Es la biblioteca de componentes de front-end más popular del mundo. Es posible realizar rápidamente un prototipo de ideas o construir una aplicación completa con variables y combinaciones SASS, sistema de cuadrícula sensible, componentes precompilados extensos y complementos potentes construidos en jQuery. [49]

En su nacimiento original se conoció el framework con el nombre de *Twitter Blueprint* debido a que en realidad nació en las propias oficinas de la compañía. Sus creadores fueron Mark Otto¹⁶ y Jacob Thornton¹⁷, que, en su constante inquietud por mejorar el entorno de desarrollo, llegaron a la conclusión de que podían aportar algo distinto e interesante al entorno de los frameworks. La intención que tenían era aumentar la consistencia que existía entre el uso de diferentes herramientas de desarrollo, algo que no siempre era habitual. Se habían encontrado con defectos, inconsistencias y problemas por la combinación de diferentes herramientas de trabajo y sabían que tenían que cambiar algo para poder continuar trabajando sin invertir grandes volúmenes de tiempo en solucionar errores y poner parches. [50]

Una tecnología como Bootstrap se instaura con solidez debido a lo bien que funciona y a los positivos resultados que es capaz de ofrecer a quienes la utilizan. Hay muchas características destacadas y ventajas que han ayudado a que esta plataforma progrese con éxito. Lo más importante es que simplifica el trabajo y lo hace sin que haya que sacrificar los resultados, la estabilidad, seguridad o

¹⁵ John Resig es un programador y empresario, conocido por ser el Evangelista de JavaScript de la Fundación Mozilla, creador y líder de desarrollo de la biblioteca JavaScript jQuery.

¹⁶ <https://twitter.com/mdo>

¹⁷ <https://twitter.com/fat>

rendimiento. Se unifican estándares y se consigue que todo sea más fluido desde el primer instante. [50]

No importa mucho qué tipo de código se esté utilizando porque al final lo que se encontrará será una navegabilidad adaptable que tendrá compatibilidad con cualquier tipo de terminal. Sin importar cual, un ordenador de sobremesa, un tablet o un *smartphone*, la visualización siempre será la óptima. Todo está integrado y es muy personalizable sin tener que contar con grandes conocimientos. También se integra con sistemas de edición y publicación como *Drupal* o *WordPress*, mientras que no es menos relevante destacar que la comunidad de profesionales que colaboran y trabajan a su alrededor es enorme [50]. Por estas razones fue escogida esta tecnología por encima de otras para el desarrollo de la aplicación web propuesta

1.4.2.6 *Mongo DB*

MongoDB (del inglés *humongous*, "enorme") es un sistema de base de datos NoSQL orientado a documentos de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, *MongoDB* guarda estructuras de datos *BSON* (una especificación similar a *JSON*) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. *MongoDB* es una base de datos adecuado para su uso en producción y con múltiples funcionalidades.

Está orientado a documentos de esquema libre, lo que implica que cada registro puede tener un esquema de datos distinto, (los atributos no tienen que repetirse entre los diferentes registros). [51]

Es una solución pensada para mejorar la escalabilidad horizontal de la capa de datos, con un desarrollo más sencillo y la posibilidad de almacenar datos con órdenes de magnitud mayores. Cuando se necesita escalar con muchas máquinas, el enfoque no-relacional resulta eficiente, y *MongoDB* una alternativa correcta. El modelo de documento de datos (*JSON/BSON*) pretende ser fácil de programar, fácil de manejar y ofrece alto rendimiento mediante la agrupación de los datos relevantes entre sí, internamente. [51]

En *MongoDB*, cada registro o conjunto de datos se denomina documento, que pueden ser agrupados en colecciones, (equivalente a las tablas de las bases de datos relacionales, pero sin estar sujetos a un esquema fijo). Se pueden crear índices para algunos atributos de los documentos. La diferencia con los SGBDR, (Sistemas Gestores de Bases de Datos Relacionales), es que no se utilizan tablas,

filas ni columnas, sino documentos con distintas estructuras. Un conjunto de Campos formará un Documento, que en caso de asociarse con otros formará una Colección. Las bases de datos estarán formadas por Colecciones, y a su vez, cada servidor puede tener tantas bases de datos como el equipo lo permita. [51]

- Entre las principales características por las que fue escogida esta tecnología por encima de otras se encuentran:
- Soporta la búsqueda por campos, consultas de rangos y expresiones regulares.
- Puede ser utilizado como un sistema de archivos, aprovechando su capacidad para el balanceo de carga y la replicación de datos en múltiples servidores.
- Tiene la capacidad de realizar consultas utilizando JavaScript del lado del servidor, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas.

1.4.2.7 Git

Git es un Sistema de Control de Versiones (VCS) diseñado por Linus Torvalds¹⁸, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente proporcionando las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida.

Desde su nacimiento en 2005, Git ha evolucionado y madurado para ser fácil de usar y aún conservar estas cualidades iniciales. Es tremadamente rápido, muy eficiente con grandes proyectos, y tiene un increíble sistema de ramificación (*branching*) para desarrollo no lineal. La principal diferencia entre Git y cualquier otro VCS es cómo Git modela sus datos. Conceptualmente, la mayoría de los demás sistemas almacenan la información como una lista de cambios en los archivos. Estos sistemas (Subversion, Perforce, Bazaar, etc.) modelan la información que almacenan como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo. Git no modela ni almacena sus datos de este modo. En cambio, Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que se confirma un cambio, o se guarda el estado de un proyecto en Git, él básicamente hace una foto del aspecto de todos los archivos en ese momento, y guarda una referencia

¹⁸ Ingeniero de software finlandés-estadounidense ganador de múltiples premios y reconocimientos en el área de las ciencias de la computación, conocido por iniciar y mantener el desarrollo del *kernel* Linux.

a esa instantánea. Para ser eficiente, si los archivos no se han modificado, Git no almacena el archivo de nuevo, sólo un enlace al archivo anterior idéntico que ya tiene almacenado. [52]

Entre las características más relevantes por las que se escogió esta tecnología se encuentran:

- Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal.
- Gestión distribuida. Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales.
- Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH.
- Gestión eficiente de proyectos de cualquier tamaño, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
- Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio (denominado autenticación criptográfica de historial).
- Los renombrados se trabajan basándose en similitudes entre ficheros, aparte de nombres de ficheros, pero no se hacen marcas explícitas de cambios de nombre con base en supuestos nombres únicos de nodos de sistema de ficheros, lo que evita posibles y desastrosas coincidencias de ficheros diferentes en un único nombre.

1.4.3 Metodologías de desarrollo de Software

La metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Existen metodologías que permiten desarrollar software de excelente calidad debido a varios aspectos como: el control que las mismas proporcionan; la posibilidad de crear las bases para el desarrollo del software y su éxito en el tiempo y costo fijados. Cada una posee características que se relacionan en mayor medida con las necesidades de una organización y las características específicas de cada proyecto a desarrollar.

Entre las metodologías más extendidas y usadas se encuentran: Proceso Unificado Racional (RUP, por sus siglas en inglés, *Rational Unified Process*), Programación extrema (XP, por sus siglas en inglés, *eXtreme Programming*) y ICONIX.

RUP constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos [53]. La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En cada ciclo de iteración, se hace exigente el uso de artefactos (un documento, un modelo, o un elemento de modelo), siendo por este motivo una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software [54].

Por otra parte, ICONIX hereda la participación y el compromiso de los usuarios finales como metodología ágil para verificar la completitud y el cumplimiento de los requisitos. No proporciona una rigurosa y extensa documentación como RUP y no puede superar la agilidad de XP. ICONIX emplea el modelo de desarrollo de software basado en el enfoque o paradigma de desarrollo de software iterativo e incremental. Utiliza el método orientado a objeto con el enfoque unificado basado en el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) [54].

A continuación, se tratar con mayor profundidad la metodología XP, la cual fue seleccionada para la modelación de la aplicación web propuesta.

1.4.3.1 La metodología de desarrollo XP

XP forma parte de un grupo de metodologías ágiles junto a SCRUM, DSDM y Cristal (ágil) de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

La metodología XP se basa en cuatro variables:

- Costo: se refiere a las máquinas, especialistas y oficinas que se necesiten en la creación de un proyecto.
- Tiempo: esta variable se refiere al tiempo de entrega del proyecto, así como al tiempo total invertido en el desarrollo de este.
- Calidad: la calidad que posea el proyecto realizado interna y externamente.

Alcance: el alcance se refiere a la intervención necesaria del cliente durante el desarrollo del software.

La metodología XP se basa en una serie de principios básicos agrupados en cuatro categorías.

- Retroalimentación a escala fina: sus principios enuncian la importancia de la planificación, las pruebas, la interacción fuerte con el cliente y la programación en equipo simultáneamente.
- Proceso continuo en lugar de por lotes: sus principios se centran en la integración continua de las versiones del código, el análisis del diseño continuo y la entrega en períodos de tiempo corto (como máximo 3 semanas).
- Entendimiento compartido: esta categoría engloba los principios referentes al diseño simple del software, la propiedad colectiva del código y el uso de un estándar de codificación que permita la legibilidad del código.
- Bienestar del programador: su único principio defiende la semana de 40 horas debido a que un programador cansado produce código de poca calidad.

1.5 Conclusiones parciales

En este capítulo se explico el proceso de AL. Se definieron los conceptos principales vinculados al problema científico y al objeto de estudio.

Mediante una revisión bibliográfica se pudo analizar el estado actual del proceso de experimentación con el framework JCLAL. Se pudo apreciar la necesidad de implementar una aplicación web que facilite la realización de experimentos utilizando el framework JCLA a través de una interfaz mas intuitiva y que fortalezca el uso del framework por más investigadores y el trabajo colaborativo.

Fueron seleccionadas las tecnologías para desarrollar la solución propuesta, las cuales son herramientas de software libre. Se utilizan lenguajes de programación como HTML, CSS, JavaScript y otras tecnologías como Node.js, Express.js, MongoDB, Bootstrap, jQuery, así como VisualStudio Code como IDE para implementar la solución propuesta. Se adoptó a XP como metodología de desarrollo de software por las características que presenta.

Capítulo 2: Diseño e implementación de la aplicación web propuesta.

En este capítulo se describen las etapas de planificación, diseño, desarrollo y pruebas de la aplicación propuesta usando la metodología ágil de desarrollo XP. Se incluye en este capítulo la descripción detallada de los Requerimientos Funcionales (RF) y los Requerimientos No Funcionales (RNF) que cumplir la solución propuesta.

2.1 Definición de requerimientos

En el proceso de desarrollo de la aplicación es un paso de vital importancia la captura de requerimientos. Estos representan las acciones que el sistema debe realizar para satisfacer las necesidades para las cuales se creó. Es por ello que su redacción es fundamental como base para la realización de futuras pruebas.

2.1.1 Requerimientos funcionales

La capacidad del sistema en cuanto a lo que puede hacer, se define en los requerimientos funcionales.

1. Configurar nuevo experimento

- 1.1. Configurar método de evaluación
- 1.2. Configurar método de muestreo
- 1.3. Configurar los datos del dataset
- 1.4. Configurar el algoritmo
- 1.5. Configurar la estrategia de consulta
- 1.6. Configurar los listeners

2. Crear una configuración dinámica de los experimentos

- 2.1. Seleccionar una configuración predeterminada

3. Guardar la configuración de un experimento

- 3.1. Guardar la configuración de un experimento en el perfil del usuario

4. Cargar la configuración de un experimento

- 4.1. Cargar la configuración de un experimento desde el perfil del usuario

5. Ejecutar experimento

- 5.1. Ejecutar un experimento usando JCLAL desde un servidor
- 5.2. Ejecutar un experimento usando JCLAL de manera local
6. **Visualizar resultados de un experimento**
 - 6.1. Visualizar de forma grafica el resultado de los experimentos
7. **Guardar historial de experimentos**
 - 7.1. Guardar los resultados de un experimento en historial de experimentación
8. **Gestionar usuarios**
 - 8.1. Crear usuario
 - 8.2. Editar usuario
 - 8.3. Borrar usuario
 - 8.4. Autenticar usuario
 - 8.5. Asignar roll del usuario
 - 8.6. Crear un equipo de usuarios

2.1.2 Requerimientos no funcionales

Los requerimientos funcionales del sistema son las propiedades, características o cualidades que debe tener el software. Estos son muy importantes para lograr la aceptación de los usuarios finales. Son los que definen cómo se verá la aplicación, dónde se usará o qué es lo que se necesita para que esto funcione. Los requerimientos no funcionales de la aplicación web son los siguientes:

1. Usabilidad.

- 1.1. El diseño debe ser lo más óptimo posible para agilizar el tiempo de respuesta del mismo.
- 1.2. La aplicación podrá ser usada por usuarios que hayan trabajado con JCLAL e igualmente los que hallan incursionado en el área de ML.

2. Rendimiento

- 2.1. La ejecución de los experimentos debe ser eficiente.

3. Soporte

- 3.1. La aplicación contará con un buen diseño que permitirá una fácil extensión.

3.2. Brindará la posibilidad de mantenimiento para ampliar sus opciones y mejorar las que ya posee.

4. Portabilidad

4.1. Las herramientas usadas para el desarrollo de la aplicación son tecnologías de *software libre*.

4.2. Como la aplicación está desarrollada con tecnologías web; ésta podrá ser usada en cualquier sistema operativo o dispositivo.

5. Ayuda y documentación en línea

5.1. Debe contar con una ayuda de forma tal que le brinde orientación al usuario respecto a las opciones con que cuenta la aplicación, utilizando ejemplos explicativos.

2.1.3 Personas relacionadas con la utilización de la aplicación web

Las personas relacionadas con la aplicación web son aquellas que obtienen algún beneficio del uso de la misma. Esta la aplicación web está orientada a investigadores en el área de AL que deseen realizar experimentos sobre conjuntos de datos utilizando el framework JCLAL de forma individual o colaborativa en entornos de red o localmente.

En el caso de la aplicación web propuesta los investigadores que deseen utilizar JCLAL para la experimentación deberán realizar la configuración de los experimentos, asumiendo el rol de usuario de forma predeterminada o con roles configurables si se desea trabajar en equipo.

2.1.4 Historias de Usuario

Las historias de usuario sustituyen a los documentos de especificación funcional. Estas historias son escritas por el propio cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Estas, deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Las historias de usuarios deben poder ser programadas entre una y cuatro semanas.

Tabla 1: Historia de Usuario No. 1 - Crear estructura de la aplicación web

Historia de usuario	
No.: 1	Nombre: Crear estructura de la aplicación web
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Bajo
Estimación: 2 semanas	Iteración asignada: 1
Descripción: Esta estructura debe servir como base para implementar todas las funcionalidades de la aplicación web posteriormente	
Observaciones: Esta estructura tendrá todos los componentes visuales sin ninguna funcionalidad.	

Tabla 2: Historia de Usuario No. 2 - Configurar nuevo experimento

Historia de usuario	
No.: 2	Nombre: Configurar un nuevo experimento.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 4 semanas	Iteración asignada: 2
Descripción: El investigador podrá configurar los distintos parámetros del experimento en la aplicación web.	
Observaciones: Configurar el experimento incluye configurar los datos del método de evaluación, el método de muestreo, el dataset, el algoritmo, la estrategia de consulta y los listeners.	

Tabla 3: Historia de Usuario No.3 - Crear una configuración dinámica de los experimentos.

Historia de usuario	
No.: 3	Nombre: Crear una configuración dinámica de los experimentos.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 2 semanas	Iteración asignada: 2
Descripción: El investigador podrá guardar los datos de un experimento que configure en la aplicación.	
Observaciones: Guardar la configuración incluye poder visualizar y exportar el archivo en formato XML.	

Tabla 4: Historia de Usuario No.4 - Gestionar usuarios.

Historia de usuario	
No.: 4	Nombre: Gestionar usuarios.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 4 semanas	Iteración asignada: 3
Descripción: El investigador podrá registrarse y crear grupos que podrá administrar.	
Observaciones: Los investigadores registrados que creen un grupo tendrán la opción de asignar roles a otros usuarios.	

Tabla 5: Historia de Usuario No.5 - Guardar la configuración de un experimento.

Historia de usuario	
No.: 5	Nombre: Guardar la configuración de un experimento.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 2 semanas	Iteración asignada: 4
Descripción: El investigador podrá guardar los datos de un experimento que configure en la aplicación.	
Observaciones: Guardar la configuración incluye poder visualizar y exportar el archivo en formato XML.	

Tabla 6: Historia de Usuario No.6 - Cargar la configuración de un experimento.

Historia de usuario	
No.: 6	Nombre: Cargar la configuración de un experimento.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 2 semanas	Iteración asignada: 4
Descripción: El investigador podrá cargar los datos de un experimento que configure en la aplicación.	
Observaciones: Cargar la configuración incluye poder importarla desde un archivo en formato XML.	

Tabla 7: Historia de Usuario No.7 - Ejecutar un experimento.

Historia de usuario	
No.: 7	Nombre: Ejecutar un experimento.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 4 semanas	Iteración asignada: 5
Descripción: El Investigador podrá ejecutar un experimento de AL usando el framework JCLAL en la aplicación.	
Observaciones: Ejecutar un experimento incluye poder hacerlo en Paralelo y de forma Distribuida usando el <i>framework</i> Spark	

Tabla 8: Historia de Usuario No.8 - Visualizar resultados de un experimento.

Historia de usuario	
No.: 8	Nombre: Visualizar resultados de un experimento.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 2 semanas	Iteración asignada: 5
Descripción: El Investigador podrá visualizar los resultados de un experimento que ejecute en la aplicación.	
Observaciones: Visualizar el experimento incluye poder ver el tiempo de ejecución, datos del experimento y una gráfica de los parámetros de este.	

Tabla 9: Historia de Usuario No.9 - Guardar historial de experimentación.

Historia de usuario	
No.: 9	Nombre: Guardar historial de experimentación.
Usuario: Todos	
Prioridad de negociación: Alta	Nivel de complejidad: Alto
Estimación: 2 semanas	Iteración asignada: 6
Descripción: El Investigador podrá guardar un historial de los resultados de los experimentos que ejecute en la aplicación.	
Observaciones: El historial podrá ser gestionado por el usuario.	

La estimación del tiempo de duración de cada historia se utilizó como unidad la semana laborable que consiste en 5 días, cada uno con 8 horas laborables. Todo esto se realiza con la práctica de la metodología utilizada que afirma que se debe trabajar como máximo 40 horas a la semana.

2.2 Planeación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. Se comienza recompilando las historias de usuario y después los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Una vez acordado el cronograma en la reunión de planificación, comienza la fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala las historias de usuario.

2.2.1 Plan de iteraciones

Para la selección del trabajo de cada iteración se tuvo en cuenta que este no tuviera más de 4 semanas de desarrollo, siguiendo las prácticas de la metodología seleccionada. En la tabla 10 se muestra la distribución de las Historias de Usuario por cada iteración.

Tabla 10: Distribución de las historias de usuario por iteración.

Distribución de las historias de usuario por iteración		
Iteraciones	Orden de las historias de usuario a implementar	Tiempo
1	Crear estructura de la aplicación web	2 semanas
2	Configurar un nuevo experimento	4 semanas
	Crear configuración dinámica de los experimentos.	
3	Gestionar usuario	4 semanas
4	Guardar la configuración de un experimento	2 semanas
	Cargar la configuración de un experimento	
5	Ejecutar un experimento	4 semanas
	Visualizar resultados de un experimento.	
6	Guardar historial de experimentación.	2 semanas

2.2.2 Plan de entregas

El plan de entregas es el compromiso final del equipo de trabajo con los clientes. Es una cuestión de vital importancia para el negocio entre ambas partes, ya que la entrega tardía o temprana de la solución, repercute notablemente en la economía y moral de todos los involucrados.

La estimación es uno de los temas complicados dentro del proceso de desarrollo de un proyecto utilizando la metodología XP y es por ello que resulta de vital importancia tener bien claros los requerimientos del cliente, el estilo de trabajo del equipo de desarrollo y el tiempo con que dispone el cliente para tener en sus manos la solución.

Iteraciones	Fecha de entrega
1ra iteración	18 de enero del 2019
2da iteración	22 de febrero del 2019
3ra iteración	22 de marzo del 2019
4ta iteración	5 de abril del 2019
5ta iteración	17 de mayo del 2019
6ta iteración	31 de mayo del 2019

2.3 Arquitectura del sistema

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología es la simplicidad y el uso de las soluciones *spike*¹⁹. El primero hace referencia a que un diseño simple se implementa más rápidamente que uno complejo, mientras que el segundo se refiere a cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario; las soluciones *spike* o programas de pruebas son utilizados para explorar diferentes soluciones; son utilizados únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

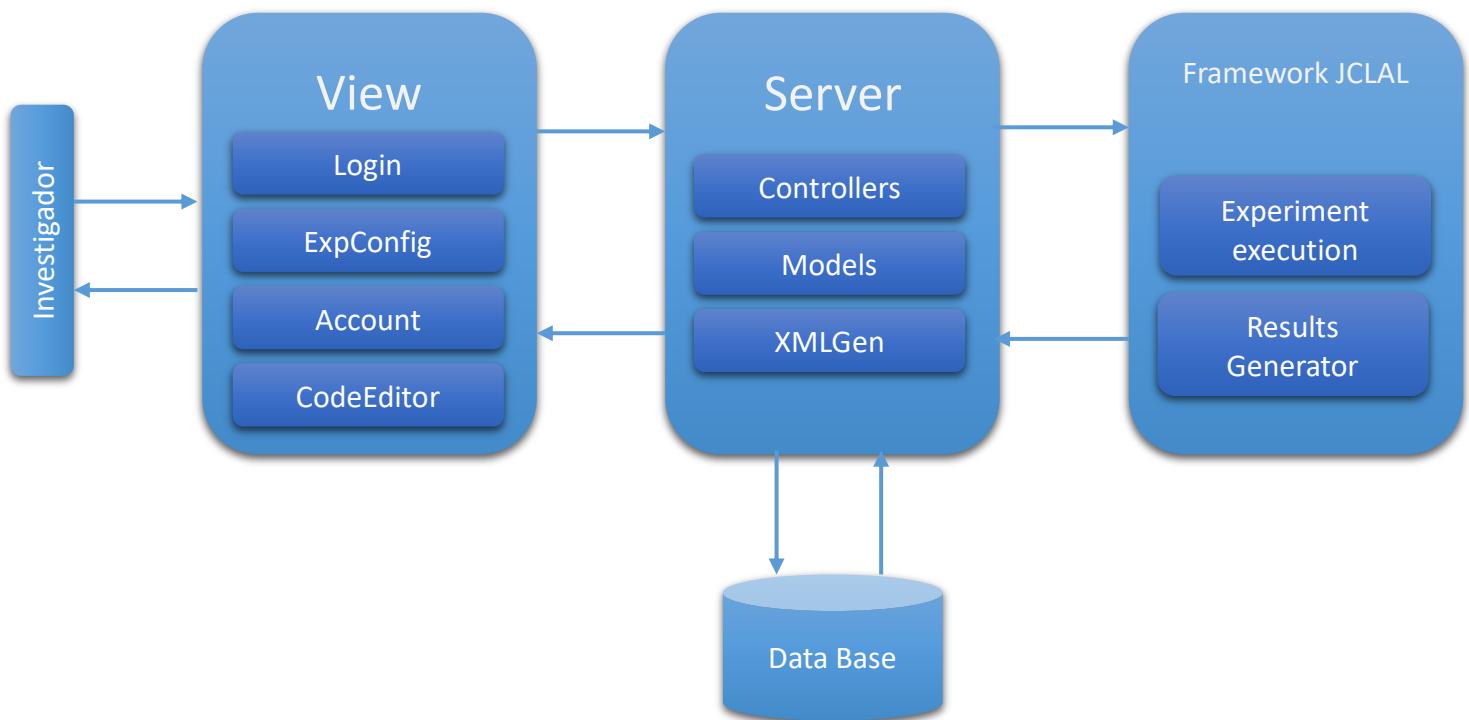


Figura 7: Arquitectura de la aplicación web

¹⁹ Spike se traduce como "punta", "clavo" o "frustrar"

2.4 Implementación

La metodología propone comenzar la implementación partiendo de una arquitectura lo más flexible posible para evitar grandes cambios en las próximas iteraciones y en los cambios que generalmente el cliente propone. Ya que la solución que se propone posee un gran componente técnico, es necesario desde un inicio tener bien definida la arquitectura. Trazada la misma, se procede a la primera iteración.

2.4.1 Iteración 1

El principal objetivo de esta iteración es crear una aplicación con la estructura necesaria para implementar el diseño estructural de la aplicación. Para ello se trazaron las siguientes tareas:

Tarea 1: Crear prototipo no funcional de la aplicación web

Tarea 2: Crear vista principal

Tarea 3: Crear vista de configuración de experimentos

Tarea 4: Crear vista de edición de código

Tarea 5: Crear vista de login

Tarea 6: Crear vista de usuario

Tabla 11: Tarea 1: Crear prototipo no funcional de la aplicación web

Tarea	
No.: 1	No. De la HU: 1
Nombre de la tarea: Crear prototipo no funcional de la aplicación web.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 7 de enero del 2019	Fecha de fin: 8 de enero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Debe estructurarse el diseño general de las diferentes vistas.	

Tabla 12: Tarea 2: Crear vista principal

Tarea	
No.: 2	No. De la HU: 1
Nombre de la tarea: Crear vista principal.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 8 de enero del 2019	Fecha de fin: 9 de enero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Debe estructurarse el diseño de la vista principal.	

Tabla 13: Tarea 3: Crear vista de configuración de experimentos

Tarea	
No.: 3	No. De la HU: 1
Nombre de la tarea: Crear vista de configuración de experimentos.	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 9 de enero del 2019	Fecha de fin: 14 de enero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Debe estructurarse el diseño de la vista donde se configurarán los nuevos experimentos.	

Tabla 14: Tarea 4: Crear vista de edición de código

Tarea	
No.: 4	No. De la HU: 1
Nombre de la tarea: Crear vista de edición de código.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 15 de enero del 2019	Fecha de fin: 16 de enero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Debe estructurarse el diseño de la vista donde se podrá editar el código.	

Tabla 15: Tarea 5: Crear vista de login

Tarea	
No.: 5	No. De la HU: 1
Nombre de la tarea: Crear vista de login.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 16 de enero del 2019	Fecha de fin: 17 de enero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Debe estructurarse el diseño de la vista donde se podrá registrar un nuevo usuario.	

Tabla 16: Tarea 6: Crear vista de usuario

Tarea	
No.: 6	No. De la HU: 1
Nombre de la tarea: Crear vista de login.	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 17 de enero del 2019	Fecha de fin: 18 de enero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Debe estructurarse el diseño de la vista donde el usuario podrá gestionar su cuenta y gestionar grupo en caso de ser organizador de uno.	

2.4.2 Iteración 2

El objetivo principal de esta iteración es implementar la configuración de nuevos experimentos, crear una configuración predeterminada que permita ejecutar un experimento de madera rápida y la configuración dinámica de un experimento basándose en las opciones que el usuario escoja en cada nivel.

Tareas:

Tarea 7: Implementar la forma en que se genera el XML.

Tarea 8: Implementar configuración predeterminada de un experimento.

Tarea 9: Implementar configuración dinámica de un experimento.

Tabla 17: Tarea 7: Implementar la forma en que se genera el XML

Tarea	
No.: 7	No. De la HU: 2
Nombre de la tarea: Implementar la forma en que se genera el XML.	
Tipo de tarea: Desarrollo	Estimación: 2 semanas
Fecha de inicio: 21 de enero del 2019	Fecha de fin: 1 de febrero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Se debe implementar la manera de generar el archivo XML que se le pasara al núcleo del framework JCLAL para la realización de un experimento.	

Tabla 18: Tarea 18: Implementar configuración predeterminada de un experimento.

Tarea	
No.: 8	No. De la HU: 2
Nombre de la tarea: Implementar configuración predeterminada de un experimento.	
Tipo de tarea: Desarrollo	Estimación: 1 semanas
Fecha de inicio: 4 de febrero del 2019	Fecha de fin: 8 de febrero del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Se debe implementar una configuración predeterminada de un experimento genérico para la ejecución rápida del mismo.	

Tabla 19: Tarea 19: Implementar configuración dinámica de un experimento.

Tarea	
No.: 9	No. De la HU: 3
Nombre de la tarea: Implementar configuración dinámica de un experimento.	
Tipo de tarea: Desarrollo	Estimación: 3 semanas
Fecha de inicio: 11 de febrero del 2019	Fecha de fin: 1 de marzo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Se debe crear la manera de ir autocompletando la configuración de un experimento de manera dinámica dependiendo de las opciones de configuración que el usuario valla seleccionando.	

2.4.3 Iteración 3

En esta iteración se implementará todo lo relacionado con la gestión de usuario; el login y toda la parte de registro de nuevos usuarios y gestión de grupos. Para lo cual se formulo la siguiente lista de tareas:

Tarea 10: Implementar formulario de registro de usuario

Tarea 11: Implementar autentificación del login.

Tarea 12: Implementar sistema de grupos de usuarios.

Tabla 20: Tarea 10: Implementar formulario de registro de usuario

Tarea	
No.: 10	No. De la HU: 4
Nombre de la tarea: Implementar formulario de registro de usuario.	
Tipo de tarea: Desarrollo	Estimación: 1 semana
Fecha de inicio: 25 de febrero del 2019	Fecha de fin: 1 de marzo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Se debe implementar el formulario para que un nuevo usuario se registre en el sistema, el cual debe contener los siguientes campos: Nombre, e-mail, contraseña, foto de perfil y roll.	

Tabla 21: Tarea 11: Implementar autentificación del login.

Tarea	
No.: 11	No. De la HU: 4
Nombre de la tarea: Implementar autentificación del login.	
Tipo de tarea: Desarrollo	Estimación: 1 semana
Fecha de inicio: 4 de marzo del 2019	Fecha de fin: 8 de marzo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar sistema de autentificación por que le permitirá al usuario iniciar sesión con su cuenta.	

Tabla 22: Tarea 12: Implementar sistema de grupos de usuarios

Tarea	
No.: 12	No. De la HU: 4
Nombre de la tarea: Implementar sistema de grupos de usuarios	
Tipo de tarea: Desarrollo	Estimación: 2 semanas
Fecha de inicio: 11 de marzo del 2019	Fecha de fin: 22 de marzo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma en la que los usuarios pueden crear grupos que permitan el trabajo de forma colaborativa mediante la asignación de roles.	

2.4.4 Iteración 4

El objetivo de esta iteración es que el sistema permita guardar y cargar la configuración de un experimento. Las tareas para esta iteración se relacionan a continuación.

Tarea 13: Implementar la forma en la que se guarda una configuración en la base de datos.

Tarea 14: Implementar la forma en la que se carga una configuración desde la base de datos.

Tabla 23: Tarea 13: Implementar la forma en la que se guarda una configuración en la base de datos

Tarea	
No.: 13	No. De la HU: 5
Nombre de la tarea: Implementar la forma en la que se guarda una configuración en la base de datos	
Tipo de tarea: Desarrollo	Estimación: 1 semanas
Fecha de inicio: 25 de marzo del 2019	Fecha de fin: 29 de marzo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma de guardar los datos de una configuración realizada por el usuario en su perfil.	

Tabla 24: Tarea 14: Implementar la forma en la que se carga una configuración desde la base de datos

Tarea	
No.: 14	No. De la HU: 6
Nombre de la tarea: Implementar la forma en la que se carga una configuración desde la base de datos	
Tipo de tarea: Desarrollo	Estimación: 1 semanas
Fecha de inicio: 1 de abril del 2019	Fecha de fin: 5 de abril del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma de cargar los datos de una configuración realizada por el usuario desde su perfil.	

2.4.5 Iteración 5

En esta iteración se implementará el modulo que permita ejecutar un experimento y visualizar los resultados de manera gráfica. Para lo cual se trazaron las siguientes tareas.

Tarea 15: Implementar la manera en la que el sistema interactúa con el *framework* JCLAL

Tarea 16: Implementar la manera de enviar el conjunto de datos al *framework* JCLAL

Tarea 17: Ejecutar un experimento usando el *framework* JCLAL

Tarea 18: Ejecutar un experimento en paralelo usando el *framework* JCLAL

Tarea 19: Ejecutar un experimento de forma distribuida usando el *framework* JCLAL

Tarea 20: Implementar la manera de procesar los datos de los resultados de un experimento

Tarea 21: Implementar la forma en la que se muestran los resultados de un experimento de forma gráfica.

Tabla 25: Tarea 15: Implementar la manera en la que el sistema interactúa con el framework JCLAL.

Tarea	
No.: 15	No. De la HU: 7
Nombre de la tarea: Implementar la manera en la que el sistema interactúa con el framework JCLAL	
Tipo de tarea: Desarrollo	Estimación: 2 semanas
Fecha de inicio: 8 de abril del 2019	Fecha de fin: 19 de abril del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Se debe implementar un modulo de comunicación entre el servidor y el framework JCLAL que permita la interacción entre estos.	

Tabla 26: Tarea 16: Implementar la manera de enviar el conjunto de datos al framework JCLAL

Tarea	
No.: 16	No. De la HU: 7
Nombre de la tarea: Implementar la manera de enviar el conjunto de datos al framework JCLAL	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 22 de abril del 2019	Fecha de fin: 23 de abril del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: El framework debe tener acceso al conjunto de datos que se va a procesar.	

Tabla 27: Tarea 17: Ejecutar un experimento usando el framework JCLAL

Tarea	
No.: 17	No. De la HU: 7
Nombre de la tarea: Ejecutar un experimento usando el framework JCLAL	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 24 de abril del 2019	Fecha de fin: 29 de abril del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma de ejecutar un experimento de manera normal usando el framework JCLAL	

Tabla 28: Tarea 18: Ejecutar un experimento en paralelo usando el framework JCLAL

Tarea	
No.: 18	No. De la HU: 7
Nombre de la tarea: Ejecutar un experimento en paralelo usando el framework JCLAL	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 30 de abril del 2019	Fecha de fin: 3 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma de ejecutar un experimento en paralelo usando el framework JCLAL	

Tabla 29: Tarea 19: Ejecutar un experimento de forma distribuida usando el framework JCLAL

Tarea	
No.: 19	No. De la HU: 7
Nombre de la tarea: Ejecutar un experimento de forma distribuida usando el framework JCLAL	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 6 de mayo del 2019	Fecha de fin: 9 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma de ejecutar un experimento de forma distribuida usando el framework JCLAL	

Tabla 30: Tarea 20: Implementar la manera de procesar los datos de los resultados de un experimento

Tarea	
No.: 20	No. De la HU: 8
Nombre de la tarea: Implementar la manera de procesar los datos de los resultados de un experimento	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 10 de mayo del 2019	Fecha de fin: 15 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Implementar la forma de procesar los datos de los resultados arrojados por el framework realizado un experimento	

Tabla 31: Tarea 21: Implementar la forma en la que se muestran los resultados de un experimento de forma gráfica

Tarea	
No.: 21	No. De la HU: 8
Nombre de la tarea: Implementar la forma en la que se muestran los resultados de un experimento de forma gráfica	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 16 de mayo del 2019	Fecha de fin: 17 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Los resultados de los experimentos deben ser mostrados de forma gráfica, comprensible para el usuario.	

2.4.6 Iteración 6

Esta iteración tiene como objetivo lograr que la aplicación almacene el historial de experimentación de cada usuario

Tarea 22: Implementar la forma de guardar los resultados en la base de datos.

Tarea 23: Implementar la manera de que el usuario pueda gestionar el historial de experimentación.

Tarea 24: Implementar la manera en que se maneja la privacidad del historial de experimentación.

Tabla 32: Tarea 22: Implementar la forma de guardar los resultados en la base de datos

Tarea	
No.: 22	No. De la HU: 9
Nombre de la tarea: Implementar la forma de guardar los resultados en la base de datos	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 20 de mayo del 2019	Fecha de fin: 23 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: Los resultados de los experimentos deben ser guardados en la base de datos para poder ser mostrados posteriormente en el historial.	

Tabla 33: Tarea 23: Implementar la manera de que el usuario pueda gestionar el historial de experimentación

Tarea	
No.: 23	No. De la HU: 9
Nombre de la tarea: Implementar la manera de que el usuario pueda gestionar el historial de experimentación	
Tipo de tarea: Desarrollo	Estimación: 4 días
Fecha de inicio: 24 de mayo del 2019	Fecha de fin: 29 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: El usuario debe poder ser capaz de gestionar su propio historial de experimentación. Acciones como modificar, eliminar o compartir deben estar incluidas.	

Tabla 34: Tarea 24: Implementar la manera en que se maneja la privacidad del historial de experimentación de cada usuario

Tarea	
No.: 24	No. De la HU: 9
Nombre de la tarea: Implementar la manera en que se maneja la privacidad del historial de experimentación	
Tipo de tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 30 de mayo del 2019	Fecha de fin: 31 de mayo del 2019
Programador responsable: Ramón Alejandro Valdés Ochoa	
Descripción: El historial debe ser privado para cada usuario y publico para los miembros del grupo si este pertenece a alguno.	

2.5 Prueba

La aplicación web fue sometida a varias pruebas en las que se observó un correcto funcionamiento y cumplimiento de los requisitos funcionales definidos con anterioridad. Los tiempos de respuestas de las diferentes funcionalidades del sistema se mantuvieron dentro de un rango razonable y la gestión de los datos se comportó de manera estable y efectiva. La interfaz tuvo una buena aceptación

debido a su sencillez y facilidad de uso, lo cual propicio, de forma general, que se obtuviera una buena Experiencia de Usuario.

2.5.1 Pruebas de Aceptación

Las pruebas de aceptación también son conocidas como pruebas de caja negra, ya que es el propio cliente quien la realiza en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones realizadas en las historias de usuario.

A continuación, aparecen las pruebas de aceptación realizadas a la solución propuesta:

Tabla 35: Caso de Prueba de Aceptación UH2-P1.

Caso de Prueba de Aceptación	
Código: HU2-P1	Historia de Usuario: 2
Nombre: Configurar un nuevo experimento.	
Descripción: Prueba la funcionalidad de crear una nueva configuración para un experimento en el framework JCLAL	
Condiciones de ejecución: Para poder crear un nuevo experimento es necesario que exista un conjunto de datos.	
Entrada/Pasos de ejecución: Acceder a la vista de configuración de un nuevo experimento desde la vista principal haciendo clic en la pestaña “Experiment Configuration”. En la vista de configuración llenar los datos necesarios para configurar un nuevo experimento. Hacer clic en el botón “Execute Experiment” luego de haber terminado.	
Resultado esperado: Poder llenar los campos sin errores y ver que la petición al servidor se realizó de manera exitosa al pulsar el botón “Execute Experiment”.	
Evaluación de la prueba: Prueba Satisfactoria	

Tabla 36: Caso de Prueba de Aceptación UH3-P2

Caso de Prueba de Aceptación	
Código: HU3-P2	Historia de Usuario: 3
Nombre: Crear configuración dinámica de los experimentos	
Descripción: El sistema configura los diferentes parámetros de la configuración dependiendo de los valores que introduzca el usuario en los diferentes campos	
Condiciones de ejecución: El usuario debe seleccionar algunos parámetros.	
Entrada/Pasos de ejecución: Acceder a la vista de configuración de un nuevo experimento desde la vista principal haciendo clic en la pestaña " <i>Experiment Configuration</i> ". En la vista de configuración parámetros necesarios para configurar un nuevo experimento.	
Resultado esperado: Poder llenar los campos sin errores y que el sistema halla seleccionado aquellos que sean correctos en función de la configuración.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 37: Caso de Prueba de Aceptación UH4-P3

Caso de Prueba de Aceptación	
Código: HU4-P3	Historia de Usuario: 4
Nombre: Gestionar usuario	
Descripción: El sistema permite una gestión eficiente de cuentas de usuario	
Condiciones de ejecución: El usuario no debe estar registrado ya en la aplicación.	
Entrada/Pasos de ejecución: Acceder a la vista de registro de nuevo usuario a través de la vista principal haciendo clic en el botón " <i>Signup</i> ". En la vista de registro de nuevo usuario llenar los campos requeridos, luego hacer clic en el botón " <i>Create</i> ".	
Resultado esperado: El usuario debe quedar registrado en el sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 38: Caso de Prueba de Aceptación UH5-P4

Caso de Prueba de Aceptación	
Código: HU5-P4	Historia de Usuario: 5
Nombre: Guardar la configuracion de un experimento	
Descripción: Prueba la funcionalidad de guardar una configuración de un experimento de JCLAL	
Condiciones de ejecución: El usuario debe iniciar sesión en la aplicación. Configurar un experimento.	
Entrada/Pasos de ejecución: En la vista de configuración de nuevo experimento, realizar la configuración de un nuevo experimento. Hacer clic en el botón "Save" en el panel "Configuración".	
Resultado esperado: El sistema mostrara un <i>alert</i> advirtiendo que la configuración ha sido guardada exitosamente	
Evaluación de la prueba: Prueba exitosa	

Tabla 39: Caso de Prueba de Aceptación UH6-P5

Caso de Prueba de Aceptación	
Código: HU6-P5	Historia de Usuario: 6
Nombre: Cargar la configuracion de un experimento	
Descripción: Prueba la funcionalidad de guardar una configuración de un experimento de JCLAL	
Condiciones de ejecución: El usuario debe iniciar sesión en la aplicación. Al menos una configuración debe haber sido guardada previamente.	
Entrada/Pasos de ejecución: En la vista de configuración de nuevo experimento, hacer clic en el botón "Load" en el panel "Configuración".	
Resultado esperado: El sistema mostrara un <i>alert</i> advirtiendo que la configuración ha sido cargada exitosamente	
Evaluación de la prueba: Prueba exitosa	

Tabla 40: Caso de Prueba de Aceptación UH7-P6

Caso de Prueba de Aceptación	
Código: HU7-P6	Historia de Usuario: 7
Nombre: Ejecutar un experimento usando el framework JCLAL	
Descripción: Prueba la funcionalidad de ejecutar un experimento de AL usando JCLAL	
Condiciones de ejecución: Todos los parámetros de configuración deben estar correctamente completados por el usuario	
Entrada/Pasos de ejecución: En la vista de configuración de nuevo experimento, realizar la configuración de un nuevo experimento. Hacer clic en el botón <i>"Run Experiment"</i> .	
Resultado esperado: El sistema mostrara una interfaz que muestra que el experimento se está ejecutando.	
Evaluación de la prueba: Prueba exitosa	

Tabla 41: Caso de Prueba de Aceptación UH8-P7

Caso de Prueba de Aceptación	
Código: HU8-P7	Historia de Usuario: 8
Nombre: Visualizar los resultados de un experimento.	
Descripción: Prueba la funcionalidad de visualizar los resultados de un experimento de AL ejecutado en el framework JCLAL	
Condiciones de ejecución: Haber ejecutado y finalizado previamente un experimento.	
Entrada/Pasos de ejecución: Luego de haber finalizado la ejecución de un experimento el sistema automáticamente mostrara los resultados.	
Resultado esperado: El sistema mostrara una <i>interfaz con los resultados del experimento</i> .	
Evaluación de la prueba: Prueba exitosa	

Tabla 42: Caso de Prueba de Aceptación UH9-P8

Caso de Prueba de Aceptación	
Código: HU9-P8	Historia de Usuario: 9
Nombre: Guardar historial de experimentación.	
Descripción: Prueba la funcionalidad de guardar un historial de experimentación.	
Condiciones de ejecución: El usuario debe iniciar sesión en la aplicación. Uno o varios experimentos deben haberse realizado.	
Entrada/Pasos de ejecución: En la vista de cuenta de usuario, acceder a la pestaña "History".	
Resultado esperado: Se mostrará el historial de experimentos realizados por el usuario y la posibilidad de visualizar los resultados.	
Evaluación de la prueba: Prueba exitosa	

2.6 Conclusiones Parciales

En el presente capítulo se abordaron todos los temas referentes a la metodología de desarrollo de software escogida en la construcción de la solución. Además, se presentó el diseño de la arquitectura de la aplicación web y las tareas que se llevaron a cabo para construirla.

La aplicación de la metodología en la implementación de la solución propuesta permitió el diseño y posterior desarrollo de la aplicación de forma rápida y organizada. Las fechas de entrega según lo planificado por el cronograma estuvieron acordes a la complejidad de desarrollo de las distintas tareas asignadas a cada iteración. El desglose de la implementación por iteraciones resultó de gran utilidad para simplificar el trabajo y organizar el período de pruebas.

La efectividad del desarrollo dirigido por pruebas y la aplicación de las pruebas de aceptación demuestran ser muy altas en el proceso de desarrollo de software. Ambas juegan un papel fundamental en el proceso de construcción del entorno de usuario con una metodología ágil, aún cuando el equipo de desarrollo es de un solo integrante como es el caso.

Conclusiones

El análisis de los fundamentos teóricos que rigen el Aprendizaje Activo y cómo se utilizan en el framework JCLAL, así como el análisis del historial de desarrollo que ha hecho evolucionar a esta herramienta como VisualJCLAL, permitieron sentar las bases de conocimientos necesarios para que con la presente investigación se cumpliera el objetivo trazado.

El uso de la metodología ágil XP para guiar el proceso, la arquitectura y el grupo de tecnologías utilizadas favorecieron el análisis, diseño e implementación de la aplicación web.

Las características y ventajas que ofrece la aplicación web logran un ahorro del tiempo y una interacción menos asistida en la experimentación usando el framework JCLAL, lo cual posibilita la aplicación y uso de JCLAL por más investigadores.

Recomendaciones

Por los resultados obtenidos en la presente investigación y para continuar el desarrollo de este trabajo se recomienda:

- Mejorar la optimización y la rapidez a la hora de realizar los experimentos
- Mejorar la interfaz de edición de código para aumentar su utilidad.
- Agregar funcionalidades que permitan una comunicación interna entre los usuarios de un grupo

Referencias bibliográficas

- [1] E. Pérez Perdomo and O. Pupo Reyes, "PROPUESTA DE UN FRAMEWORK PARA," Universidad de Holguín "Oscar Lucero Moya," 2013.
- [2] B. Settles, "Active learning literature survey," *Univ. Wisconsin, Madison*, vol. 15, no. 2, pp. 201–221, 2010.
- [3] O. Reyes, E. Pérez, M. Del, C. Rodríguez-Hernández, H. M. Fardoun, and S. Ventura, "JCLAL: A Java Framework for Active Learning," *J. Mach. Learn. Res.*, vol. 17, pp. 1–5, 2016.
- [4] L. M. Sánchez Velázquez, "VISUALJCLAL: ENTORNO DE USUARIO PARA EL FRAMEWORK JCLAL," Universidad de Holguín "Oscar Lucero Moya," 2017.
- [5] N. J. Nilsson, "Introduction to Machine Learning: An Early Draft of a Proposed Textbook.," p. 179, 1996.
- [6] T. Mitchell, *Machine Learning*. 1997.
- [7] S. Huang, W. Gao, and Z. Zhou, "Fast Multi-Instance Multi-Label Learning," vol. 176, no. 1, pp. 2291–2320, 2012, 2012.
- [8] S. Huang, N. Gao, and S. Chen, "Multi-Instance Multi-Label Active Learning," 2006, no. 61503182, pp. 1886–1892.
- [9] R. Liere and P. Tadepalli, "Active learning with committees for text categorization," *Proc. Fourteenth Natl. Conf. Artif. Intell.*, pp. 591–596, 1997.
- [10] Y. Zhang, W. Xu, and J. Callan, "Exploration and Exploitation in Adaptive Filtering Based on Bayesian Active Learning," *Int. Conf. Mach. Learn.*, pp. 896–903, 2003.
- [11] A. Culotta, T. Kristjansson, A. McCallum, and P. Viola, "Corrective feedback and persistent learning for information extraction," *Artif. Intell.*, vol. 170, no. 14–15, pp. 1101–1122, 2006.
- [12] M. Becker, B. Hachey, B. Alex, and C. Grover, "Optimising selective sampling for bootstrapping named entity recognition," *ICML2005 Work.*, pp. 5–11, 2005.
- [13] S. Argamon-engelson, "Committee-Based Sample Selection For Probabilistic Classifiers," vol. 11, pp. 335–360, 1999.
- [14] E. Ringger *et al.*, "Active Learning for Part-of-Speech Tagging: Accelerating Corpus Annotation," *LAW '07 Proc. Linguist. Annot. Work. June, 2007, Prague*, no. June, pp. 101–108, 2007.
- [15] M. Becker and M. Osborne, "A two-stage method for active learning of statistical grammars," *IJCAI Int. Jt. Conf. Artif. Intell.*, no. January 2005, pp. 991–996, 2005.
- [16] B. Hachey, B. Alex, and M. Becker, "Investigating the effects of selective sampling

- on the annotation task," p. 144, 2010.
- [17] S. Zhao and H. T. Ng, "Domain Adaptation with Active Learning for Coreference Resolution," no. June, pp. 21–29, 2015.
 - [18] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Commun.*, vol. 45, no. 2, pp. 171–186, 2005.
 - [19] J.-S. Kuo, H. Li, and Y.-K. Yang, "Learning transliteration lexicons from the web," no. July, pp. 1129–1136, 2007.
 - [20] M. Sassano, "An empirical study of active learning with support vector machines for Japanese word segmentation," no. July, p. 505, 2007.
 - [21] D. Angluin, "Queries and Concept Learning," *Mach. Learn.*, vol. 53, no. 1983, pp. 1689–1699, 2013.
 - [22] D. A. Cohn, "Neural network exploration using optimal experiment design," *Neural Networks*, vol. 9, no. 6, pp. 1071–1083, 1996.
 - [23] D. D. Lewis, "A sequential algorithm for training text classifiers," *ACM SIGIR Forum*, vol. 29, no. 2. pp. 13–19, 2007.
 - [24] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," *Proc. ninth ACM Int. Conf. Multimed. - Multimed. '01*, no. C, p. 107, 2001.
 - [25] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification Simon," *CrossRef List. Deleted DOIs*, vol. 1, pp. 45–66, 2000.
 - [26] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Semisupervised SVM batch mode active learning with applications to image retrieval," *ACM Trans. Inf. Syst.*, vol. 27, no. 3, pp. 1–29, 2009.
 - [27] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," p. 1070, 2010.
 - [28] C. A. Thompson, V. Hall, and R. J. Mooney, "Active Learning for Natural Language Parsing and Information Extraction LEARNING SYSTEMS," *Mach. Learn.*, no. June, pp. 406–414, 1999.
 - [29] D. Hakkani-Tur, G. Riccardi, and A. Gorin, "Active learning for automatic speech recognition," no. December 2016, p. IV-3904-IV-3907, 2011.
 - [30] S. Huang, C. A. I. Nianguang, P. Penzuti Pacheco, S. Narandes, Y. Wang, and X. U. Wayne, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics and Proteomics*, vol. 15, no. 1, pp. 41–51, 2018.
 - [31] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," 2004, pp.

287–294.

- [32] Y. Freund, H. S. SEUNG, E. Shamir, and N. Tishby, "Selective Sampling Using the Query by Committee," *Mach. Learn.*, vol. 168, pp. 133–168, 1997.
- [33] D. J. C. MacKay, "Information based objective functions for active data selection.pdf." .
- [34] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Herves, "JCLEC: A Java Framework for Evolutionary Computation," *Soft Comput.*, vol. 12, no. 4, pp. 381–392, Oct. 2007.
- [35] G. Tsoumakas and J. Vilcek, "MULAN : A Java Library for Multi-Label Learning," vol. 12, pp. 2411–2414, 2011.
- [36] S. Luján-Mora, "Programación de aplicaciones web: historia, principios básicos y clientes web," Oct. 2002.
- [37] World Wide Web Consortium, "<https://www.w3.org/html/>." [Online]. Available: <https://www.w3.org/html/>.
- [38] mozilla, "CSS: Cascading Style Sheets | MDN." [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>. [Accessed: 07-Mar-2019].
- [39] E. International, "ECMA-262," 2009.
- [40] David Flanagan, "JavaScript : The Definitive Guide , 4th Edition By David Flanagan Lilmeanman," 2004.
- [41] Oracle, "No Title." [Online]. Available: <https://www.oracle.com/technetwork/java/javase/12u-relnotes-5211424.html>. [Accessed: 08-Mar-2019].
- [42] Oracle, "Sun & Oracle." [Online]. Available: <http://www.oracle.com/us/sun/index.html>. [Accessed: 08-Mar-2019].
- [43] A. SILBERSCHATZ, U. de Yale, H. F. KORTH, U. de Lehigh, S. SUDARSHAN, and B. Instituto tecnológico indio, *Fundamentos de Bases de Datos 5a Edicion.pdf*. 2006.
- [44] Microsoft, "<https://code.visualstudio.com/docs/editor/whyscode>." .
- [45] N. j. Foundation, "Node.js." [Online]. Available: <https://nodejs.org/es/about/>. [Accessed: 28-Jan-2019].
- [46] M. Web Docs, "Introducción a Express/Node." [Online]. Available: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction. [Accessed: 28-Jan-2019].
- [47] T. jQuery Foundation, "JQuery." [Online]. Available: <http://jquery.com>. [Accessed: 28-Jan-2019].

- [48] J. Chaffer and K. Swedberg, *Learning jQuery - Fourth Edition*. 2013.
- [49] "No Title." [Online]. Available: <http://getbootstrap.com/>. [Accessed: 29-Jan-2019].
- [50] Universitat de Barcelona, "Bootstrap ¿que es y para qué sirve?" [Online]. Available: <https://www.obs-edu.com/int/blog-investigacion/sistemas/bootstrap-que-es-y-para-que-sirve>. [Accessed: 29-Jan-2019].
- [51] S. Bellido Sanchez, "e-Assessment mediante Bases de Datos NoSQL," Universidad de Sevilla.
- [52] S. Chacon, "Git --local-branching-on-the-cheap." [Online]. Available: <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>. [Accessed: 29-Jan-2019].
- [53] I. Jacobson, G. Booch, and J. Rumbaugh, *El proceso unificado de desarrollo de software*. Pearson Educación, 2000.
- [54] D. Rosenberg and M. Stephens, *Use Case Driven Object Modeling with UML Theory and Practice*. .