



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

**“IMPLEMENTACIÓN DE UNA APLICACIÓN WEB SOCIAL PARA
EL ETIQUETADO DE FRAGMENTOS DE VIDEOS DE YOUTUBE
APLICANDO EL CONCEPTO DE ‘JUEGOS CON UN
PROPÓSITO’.”**

TESIS DE GRADO

Previo la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

Presentada por:

FÉLIX MARCIAL RIVAS TOMALÁ

CARLA ANDREA SALVATIERRA ARELLANO

GUAYAQUIL – ECUADOR
2013

AGRADECIMIENTO

Agradecemos a Dios, por tantos favores recibidos, por darnos fuerza para continuar. A nuestros padres que nos han guiado siempre con sus consejos y preocupación diaria para que todo en nuestra formación salga de la mejor manera. A nuestros maestros por guiarnos en nuestra vida profesional, de manera especial a la Ing. Carmen Vaca por ser una mentora incondicional, por la confianza que ha depositado en nosotros desde el inicio. A nuestro director por el tiempo dedicado para que todo el proceso salga adelante. A nuestros amigos y compañeros que de alguna manera también nos ayudaron a concluir esta etapa.

DEDICATORIA

Dedico con mucho cariño este trabajo, A Dios por guiar mis pasos y haberme permitido alcanzar esta meta. A mis padres por su apoyo incondicional y por ser mi ejemplo a seguir. A mis hermanos por ser una fuente de inspiración para siempre salir adelante. A mi compañero de tesis porque sin el equipo formado no hubiera sido posible culminar con éxito esta etapa de formación profesional.

Carla Salvatierra Arellano

DEDICATORIA

A Dios por darme la oportunidad de vivir en este mundo y aprender. A mi madre por su incondicional amor, sus consejos y sacrificio. A mi padre por ser un ejemplo y apoyo. A mis hermanas y familiares por la compañía y palabras de apoyo en todo momento. A mi compañera de tesis por su dedicación y esmero durante todo el proceso.

Félix Rivas Tomalá

DECLARATORIA EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**”

(Reglamento de Graduación de la ESPOL)

FÉLIX MARCIAL RIVAS TOMALÁ

CARLA ANDREA SALVATIERRA ARELLANO

TRIBUNAL DE SUSTENTACIÓN

Dr. Boris Vintimilla Ph.D

PRESIDENTE

Ing. Jorge Rodríguez.

DIRECTOR DE TESIS

Dr. Sixto García Ph.D.

MIEMBRO PRINCIPAL

RESUMEN

El presente trabajo presenta el diseño e implementación de una aplicación Web en cuyo modelo se ha aplicado el concepto de “Juegos con un propósito” y utilizado “Input-Agreement” para generar descripciones de texto de manera automática y a partir de material audiovisual. Dicha aplicación tiene como objetivo conseguir la mayor cantidad de descripciones válidas para fragmentos de vídeo. El modelo propuesto es agnóstico con respecto al tipo de video que está siendo anotado. La diferencia principal con otros frameworks para anotación de material multimedia es que el juego colecta etiquetas asociadas a marcas de tiempo con una fracción del costo que requeriría esta operación al contratar usuarios expertos. El dataset sobre el cual se realizaron las pruebas está conformado por archivos almacenados en YouTube y específicamente videos del canal de ESPOL.

Para mostrar la utilidad de la aplicación desarrollada se han recolectado datos con un restringido grupo de usuarios y vídeos. Dichos usuarios generan etiquetas para distintos tipos de vídeo que posteriormente son sometidas a un proceso de filtrado de información. Como resultado de las pruebas, se observa

un conjunto de anotaciones que, evaluadas por un ser humano, resultan relevantes al contexto del video. Esto permite comprobar la efectividad del paradigma denominado “Human Computation”. Una última importante aportación del trabajo expuesto se refiere al hecho de que el modelo desarrollado puede ser utilizado para la generación de etiquetas asociadas a videos en cualquier idioma. En particular, las pruebas, generadas produjeron etiquetas para videos en español.

ÍNDICE GENERAL

RESUMEN.....	vi
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xi
ABREVIATURAS	xii
INTRODUCCIÓN.....	xiii
CAPÍTULO 1	16
ANÁLISIS DEL PROBLEMA.....	16
1.1. Antecedentes.....	16
1.2. Descripción del problema.....	17
1.3. Justificación	18
1.4. Propuesta de solución.....	19
1.5. Alcance.....	22
CAPÍTULO 2	23
FUNDAMENTOS TEÓRICOS	23
2.1. Human Computation	23
2.1.1. Definición.....	24
2.1.2. Áreas relacionadas.....	26
2.1.3. Crowdsourcing.....	28
2.2. Juegos con un propósito (Games with a purpose)	30
2.2.1. Definición y modelos	31
2.2.2. Aplicaciones	34
2.2.3. Calidad de las etiquetas	36
2.2.4. Estrategias para evitar acuerdos entre jugadores.....	38
2.2.5. Incentivos	40
CAPÍTULO 3	42
ANÁLISIS Y DISEÑO DE LA APLICACIÓN	42
3.1. Análisis de la solución.....	42
3.1.1. Requisitos funcionales.....	43
3.1.2. Requisitos no funcionales.....	45
3.2. Diseño de la solución	45
3.2.1. Diseño Lógico de la aplicación	48
3.2.2. Interfaz de usuario.....	51
3.2.3. Filtrado de la información	55
3.2.4. Diseño de plan de pruebas.....	56

CAPÍTULO 4	65
IMPLEMENTACIÓN Y PRUEBAS	65
4.1. Plataforma tecnológica.....	65
4.1.1 YouTube.....	65
4.1.2. Facebook.....	66
4.1.3. Node.js y Socket.io (WebSockets).....	67
4.2. Arquitectura de la aplicación	69
4.3. Interfaz de usuario	72
4.4. Calidad de etiquetas	74
4.5. Pruebas	75
4.5.1. Pruebas de funcionalidad.....	75
4.5.2. Calidad de la información recolectada	80
4.5.3. Análisis de resultados	82
CONCLUSIONES	86
RECOMENDACIONES	88
ANEXOS	91
ANEXO 1	92
ANEXO 2	93
Manual de Instalación.....	93
Pruebas de la aplicación.....	102
BIBLIOGRAFÍA	107

ÍNDICE DE FIGURAS

Figura 2. 1 - Áreas relacionadas	27
Figura 3. 1 - Diseño lógico de 'CazaVideos'.....	49
Figura 3. 2 - Mecanismo de "Input-Agreement" aplicado en 'CazaVideos'	51
Figura 3. 3 - Interfaz principal del juego	52
Figura 3. 4 - Interfaz de Game Recap.....	53
Figura 3. 5 - Interfaz de Ranking o Puntajes	54
Figura 3. 6 - Diseño cuasi-experimental.....	57
Figura 4. 1 – Arquitectura de la aplicación.	70
Figura 4. 2 - Intercambio de mensajes.....	71
Figura 4. 3 - Menú Principal del Juego.....	72
Figura 4. 4 - Instrucciones del Juego.	73
Figura 4. 5 - Interfaz del Juego.	74
Figura 4. 6 - Inicio de Nueva Partida.....	76
Figura 4. 7 - Obtención de Anotaciones durante el Juego.....	77
Figura 4. 8 - Resumen del Juego.	78
Figura 4. 9 - Módulo de Node.js - Socket.io	79
Figura 4. 10 - Fragmento de video entre 00:11:50 y 00:12:20.....	81
Figura 4. 11 - Etiquetas antes del filtrado.....	81
Figura 4. 12 - Etiquetas después del filtrado	82

ÍNDICE DE TABLAS

Tabla I - Juegos de Human Computation.....	34
Tabla II – Tabla Comparativa de Polling, Long Polling y WebSockets	47
Tabla III - Características incluidas en las pruebas de funcionalidad	60
Tabla IV - Métricas incluidas en las pruebas de funcionalidad	62
Tabla V - Caso de Prueba 1 – Sincronización de fragmentos de video.....	62
Tabla VI - Caso de Prueba 2 – Sincronización de mensajes.....	63
Tabla VII - Caso de Prueba 3 – Acción de decisión “Igual o diferente”.....	64
Tabla VIII – Estadísticas del Juego.	83
Tabla IX – Etiquetas válidas.	84
Tabla X – Cantidad de etiquetas bloqueadas del intervalo 12.....	85
Tabla XI – Cantidad de etiquetas válidas por usuario del intervalo 12	85

ABREVIATURAS

API	Application Programming Interface
AJAX	Asynchronous JavaScript and XML
ESPOL	Escuela Superior Politécnica Del Litoral
GWAP	Games with a Purpose
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
JSON	JavaScript Object Notation
PHP	PHP Hypertext Preprocessor
TCP	Transmission Control Protocol
W3C	World Wide Web Consortium

INTRODUCCIÓN

El avance de la tecnología ha permitido, que hoy en día la solución a muchos problemas sea obtenida por un algoritmo corriendo en un computador. Sin embargo, existen aún problemas que sólo los humanos pueden resolver. Estos problemas se refieren especialmente a tareas que requieren de la creatividad y percepción humana para su solución, como la generación de descripciones de texto para diferentes marcas de tiempo en un video. Modelos de aplicaciones para combinar aportaciones de humanos y de algoritmos han sido propuestas en el área de Human Computation [1]. Otro concepto importante de notar es uno de los propuestos en mecanismos de Crowdsourcing, el cual afirma que la inteligencia colectiva de grupos de personas genera mejores resultados que los de un individuo experto [2]. Delegando diferentes partes de la tarea a los humanos y combinando la inteligencia colectiva de un grupo de participantes, la tarea puede ser fácilmente automatizada.

Dentro de esta misma área se ha desarrollado el concepto de “Juegos con un propósito” o GWAP (Games with a purpose) por sus siglas en inglés. Los GWAP buscan aprovechar el gran número de personas que juegan en línea para

resolver problemas donde cada participante, mientras se divierte, contribuye sin saberlo a realizar una parte del procesamiento de la solución.

Hay 3 mecanismos o plantillas básicas que definen la estructura de un GWAP, y son: Output-Agreement, Input-Agreement e Inversion-Problem. De los cuales, aquel, ideal para obtener información sobre recursos de alto contenido descriptivo como videos, sonidos y música, es el llamado “Input-Agreement”, que consiste en escoger parejas de jugadores y darles entradas iguales o diferentes, para que, permitiéndoles la comunicación entre ellos, puedan intercambiar descripciones de las entradas dadas y determinar si son iguales o no [3].

Desde el primer GWAP propuesto por von Ahn denominado ESP [4], se han desarrollado varias aplicaciones como Peekaboom [5], Verbosity [6] y TagATune [7], que permiten tanto el etiquetado de imágenes, como de archivos de audio. Aun así, en la actualidad, no existe una alternativa para la generación automática de descripciones de videos, asociadas a intervalos de tiempo predefinidos.

Aplicando estos conceptos y con el objetivo de contribuir al etiquetado de recursos educativos, esta investigación busca diseñar una aplicación Web que pueda utilizar el poder de atracción de los juegos en línea para obtener etiquetas referentes al contexto desarrollado en un determinado fragmento de vídeo.

CAPÍTULO 1

ANÁLISIS DEL PROBLEMA

1.1. Antecedentes

Los conceptos relacionados a GWAP han sido establecidos por investigadores desde el año 2005 en el área de Human Computation, la misma que hasta la fecha ha crecido en volumen y calidad de publicaciones.

Los GWAP ya han sido extensamente probados como métodos de generación de metadatos en idioma inglés, que incluyen etiquetas de imágenes [8], anotaciones de contenido [9] y anotaciones de fragmentos de audio [7]. El reto principal de este tipo de juegos, consiste en diseñarlos de tal forma que motive y mantenga la atención de los usuarios, asegurándose que al mismo tiempo se genere información válida.

Las líneas de investigación paralelas a la presente, datan desde el 2005 donde se desarrollaron varios GWAP, como por ejemplo, el juego ESP; Proyecto que logró recaudar 1.3 millones etiquetas en menos de tres meses, atrayendo a 13,500 usuarios [8] y TagATune [7], el cual después de siete meses de estar en línea consiguió un total de 30,237 fragmentos de audio etiquetados, 108,558 etiquetas verificadas (por lo menos dos coincidencias) y 70,908 etiquetas únicas, generando un promedio de cuatro etiquetas por minuto de juego [10].

1.2. Descripción del problema

Actualmente el mayor repositorio de videos del mundo es “YouTube”, la plataforma administrada por Google. Esta inmensa colección de videos incluye recursos que podrían ser utilizados en educación y contribuir al desarrollo cultural de audiencias diversas. Dichos videos pertenecen a un extenso rango de categorías: desde presentaciones de estudiantes, hasta conferencias magistrales. Sin embargo, en un alto porcentaje de los casos, la descripción textual no incluye marcas de tiempo o ‘timestamps’ que guíen al usuario en el proceso de selección del fragmento del video más relevante para el tema que quiere aprender. Los estudiantes que

necesitan información sobre temas específicos deben invertir mucho tiempo en encontrar el lapso en el que se desarrolla dicho argumento.

Debido a la forma exponencial en la que crece el número de videos publicado en YouTube, sería imposible contratar expertos para asignar etiquetas a los intervalos en los que se desarrolla cada subtema. El costo de tal actividad sería notablemente alto, pues se requiere conocimiento del lenguaje y del tema expuesto.

1.3. Justificación

Considerando que el material audiovisual publicado en YouTube puede ser re-utilizado no solamente como una fuente de información, sino también como un medio de capacitación o enriquecimiento cultural y teniendo en cuenta que el creciente número del mismo, se da cada minuto en la plataforma, a los usuarios se les dificulta la búsqueda de videos y aquellos de buena calidad tal vez no son descubiertos.

Por ello, contar con descripciones para cada intervalo que se pueden considerar como subtemas del concepto principal podría incentivar a los usuarios a hacer búsquedas de una mayor cantidad de vídeos. Así mismo la recuperación de información multimedia arroja resultados mucho más

precisos cuando existen anotaciones de texto asociadas. Un modelo de etiquetado de vídeos de esta forma, podría ser utilizado en el futuro para construir una aplicación que produzca texto resumiendo el contenido del video para diferentes momentos en el tiempo.

1.4. Propuesta de solución

Este trabajo propone el diseño, implementación y análisis de un juego en línea como una aplicación Web, por medio de la cual, sus usuarios puedan como objetivo general, producir etiquetas para fragmentos específicos de vídeo.

Se han identificado unos objetivos específicos que guiarán la implementación de la aplicación, estos objetivos se presentan a continuación:

Objetivos Específicos:

1. Diseñar un escenario en el cual se aplique el concepto de “Input-Agreement” en un juego cuyo propósito es la obtención de anotaciones para material audiovisual.

2. Diseñar las reglas de juego y el algoritmo para asignar fragmentos de vídeo a parejas de jugadores de tal modo que se obtenga cierta redundancia en las etiquetas asignadas a cada ítem.

3.- Implementar “CazaVideos” como una aplicación social para la plataforma Facebook.

4.- Analizar la información recogida durante el tiempo de vida de la aplicación para evaluar la eficacia de la aplicación en cuanto a volumen y calidad de las anotaciones generadas por los jugadores.

Metodología

Se realizará una revisión del estado del arte, en la que se buscarán diferentes mecanismos que faciliten la recolección de etiquetas masivamente, así como métodos que ayuden al proceso de filtrado de la información más relevante.

Se utilizarán mecanismos diseñados por investigadores en el área de Human Computation para crear una aplicación social en la que los usuarios como un resultado colateral generen etiquetas para material

multimedia seleccionado previamente. Una vez que la aplicación se ha publicado, se observará el patrón de comportamiento de los jugadores al interactuar de manera colaborativa. Y finalmente se realizará un procesamiento offline en el que se usarán técnicas estándares de procesamiento de texto como la remoción de las “Palabras Vacías”¹ previo a la obtención de las etiquetas finales.

Resultados esperados

Se analizarán las interacciones de juego y los metadatos para generar un conjunto de métricas, entre las que están:

1. Nivel de redundancia: ¿Cuántos usuarios se necesitan en una aplicación como la propuesta para tener un conjunto de etiquetas de calidad para un intervalo de vídeo?
2. ¿Cuántas sesiones de juego se requieren para generar este conjunto de datos?

¹ “Palabras Vacías” o “Stop Words” en inglés: términos comunes, como preposiciones.

3. Volumen de metadatos generado: ¿Qué volumen de metadatos es posible generar en un período específico de tiempo utilizando GWAP en el escenario propuesto?

1.5. Alcance

La aplicación utilizará intervalos de tiempo, de vídeos previamente seleccionados. Estos constituirán el conjunto de datos en el cual se evaluará la eficacia del sistema de colección de anotaciones.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

La idea de los GWAP está fuertemente ligada a problemas de Inteligencia Artificial que aún presentan dificultades para ser automatizados por el computador. En este capítulo se explican los fundamentos teóricos detrás de los conceptos de Human Computation y GWAP.

2.1. Human Computation

En 1950, Alan Turing escribió: "La idea detrás de los ordenadores digitales se puede explicar diciendo que estas máquinas tienen la intención de llevar a cabo cualquier operación que pueda ser realizada por un equipo humano" [11].

David Alan Grier, en el 2005, escribe en su libro "When Computers were Human" [12], que antes de las Palms, iPods, PCs y Laptops, se conocía

como “computadoras” a personas que resolvían cálculos científicos a mano.

Ambos autores revelan dos formas de Computación, una basada sólo en computadoras y la otra basada en humanos. Human Computation combina estas dos formas involucrando a humanos dentro del proceso computacional.

2.1.1. Definición

Human Computation es un paradigma que utiliza el poder de procesamiento del cerebro humano, aprovechando sus destrezas y habilidades de manera original, para resolver problemas que las computadoras no han podido resolver [1]. Otra definición que se puede citar es la de Yang et al. [13] quienes definen a Human Computation como el proceso computacional que involucra a los humanos en ciertos pasos.

Human Computation es un área de investigación que se ha aplicado en la resolución de varios problemas, donde los humanos se vuelven

responsables del cómputo. Según Yuen et al. [14], los sistemas de Human Computation se han clasificado en:

Sistemas de Human Computation de iniciación.- Fueron los primeros sistemas propuestos, para aprovechar destrezas propias del ser humano. Ejemplos: Captcha y ReCaptcha [1] que inducen al usuario a reconocer palabras, cuya forma aún no puede ser reconocida fácilmente por los algoritmos existentes.

Sistemas de Human Computation distribuidos.- Fueron propuestos para extraer información útil a través de la contribución del alto número de usuarios de Internet. Ejemplos de estos, son: Wikipedia, Yahoo! Answers y MajorMiner [15]. Este último es un juego para etiquetado de música, que evalúa en modo fuera de línea, o también llamado “offline”, las etiquetas ingresadas contra aquellas ya ingresadas en la base de datos.

Quinn y Bederson [16] no consideran a Wikipedia como un sistema de Human Computation, ellos aclaran que el sitio no fue diseñado para reemplazar la tarea de una computadora sino que nació como un

proyecto de escritura colaborativa, en reemplazo a los antiguos escritores profesionales de enciclopedia.

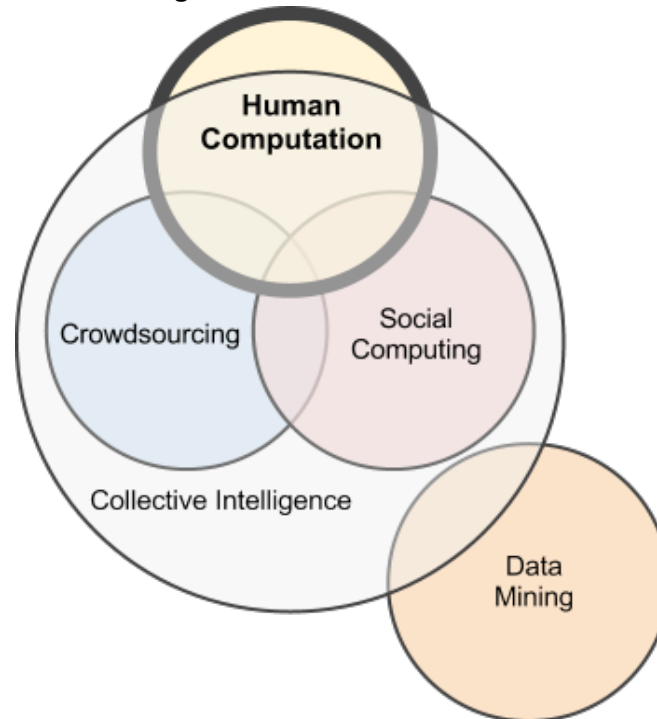
Sistemas de Human Computation sociales, basados en juegos.-

Fueron propuestos para motivar a la gran cantidad de jugadores en línea a proveer información correcta para resolver problemas difíciles de Inteligencia Artificial, como efecto colateral de su actividad. Ejemplos: el ESP game y TagATune.

2.1.2. Áreas relacionadas

Human Computation se relaciona con áreas como: Inteligencia Colectiva (Collective Intelligence), Crowdsourcing y Computación Social (Social Computing).

Figura 2. 1 - Áreas relacionadas



Fuente: Quinn A. y Bederson B. Human computation: A survey and taxonomy of a growing field. In Proceedings of CHI 2011.

Quinn y Bederson [16] explican de manera ilustrativa (ver Figura 2.1) e indican de esta manera, las diferencias entre todas esas áreas, que suelen confundirse con Human Computation:

Inteligencia Colectiva se refiere a la idea general de que grupos trabajando juntos pueden resolver problemas complejos. Human Computation se separa un poco de ésta área porque a pesar de que ambos hablan de una multitud involucrada, la Inteligencia Colectiva se

aplica sólo cuando hay un grupo de por medio, mientras que en algunos casos y muy raros, Human Computation puede aplicarse incluso a pocos o una persona realizando una tarea de forma aislada.

Minería de Datos o Data Mining es la aplicación de algoritmos específicos para la extracción de los datos a partir de patrones. Minería de Datos excluye al rol humano, por eso no se interseca con Human Computation, Crowdsourcing o Computación Social.

Computación Social se distingue de Human Computation porque estudia o facilita el comportamiento natural de humanos mediante la tecnología, mientras que la participación en Human Computation es dirigida por el proceso computacional en sí. Ejemplos de Social Computing son: las tecnologías como los blogs, wikis y comunidades en línea.

2.1.3. Crowdsourcing

Consiste en externalizar tareas tradicionalmente desarrolladas por empleados o contratistas, a un grupo masivo de personas en una llamada abierta [17]. Crowdsourcing impulsa esta colaboración masiva para lograr objetivos de negocio, pero que implican beneficios mutuos,

pues el trabajo de las masas es recompensado mediante dinero, premios o por reconocimientos.

Quinn y Bederson [16] aclaran que Crowdsourcing y Human Computation no son lo mismo. El primero reemplaza el trabajo de empleados tradicionales por un grupo masivo de personas (público en general), mientras que Human Computation reemplaza a las computadoras por la participación humana, pero manteniendo un modelo computacional incluyendo conceptos como sistema y algoritmo en el que intervienen entradas, proceso y salidas.

Quinn y Bederson indican también que la parte coincidente entre estas dos áreas es aquella representada por aplicaciones que podrían ser resueltas igualmente por ambos roles: humanos tradicionales o el computador. La traducción es un ejemplo de estas aplicaciones, pues es una tarea que puede ser resuelta por computadoras si se requiere una solución rápida o por humanos si lo que se requiere es calidad. O se puede hacer una combinación de ambos.

2.2. Juegos con un propósito (Games with a purpose)

Una de las formas de motivar a las personas a dedicar su valioso tiempo construyendo la Web Semántica es a través de diferentes incentivos [8].

A diferencia de los procesadores, los seres humanos requieren algún tipo de motivación para ser parte de una computación masiva. Los juegos en línea son un método atractivo para animar a la gente a participar en este proceso. Estos juegos constituyen un mecanismo general para utilizar el poder del cerebro humano en la resolución de problemas abiertos [1].

El diseño de estos juegos, es similar al diseño de algoritmos, se debe probar que sea correcto, entretenido y eficiente, en otras palabras, que la mecánica del juego resuelva correctamente instancias del problema y que a la vez resulte entretenido. Así mismo su eficiencia puede ser analizada, porque juegos más eficientes pueden sustituir a los menos eficientes. La única diferencia es que, en lugar de utilizar un procesador de silicio, estos “algoritmos” corren en un procesador que consiste en humanos ordinarios interactuando a través de Internet [1].

2.2.1. Definición y modelos

Los GWAP son una clase de sistemas de Human Computation, capaces de obtener información útil a través de mecanismos que involucren a los juegos.

Hay tres plantillas básicas o mecanismos que describen la estructura de un GWAP [3]:

Juegos basados en el mecanismo Output-Agreement

Configuración inicial: Se escoge a 2 jugadores aleatoriamente entre los disponibles.

Reglas: Se les da la misma entrada y deben producir resultados basados en ella tratando de producir la misma respuesta de sus compañeros. Los jugadores no comparten sus respuestas, ni pueden comunicarse entre ellos.

Condición para ganar: Deben lograr “pensar como el otro”, es decir, deben coincidir en sus respuestas, no al mismo tiempo, pero en algún momento coincidir. Esto parcialmente verifica que las salidas sean correctas.

Juegos basados en el mecanismo Input-Agreement

El Input-Agreement es una generalización del juego TagATune de Edith Law:

Configuración inicial: Se escoge a 2 jugadores aleatoriamente entre los disponibles.

Reglas: Se les da una entrada conocida por el juego que puede ser “Igual o Diferente” y deben producir resultados que describan las entradas dadas, de tal manera que sus compañeros puedan adivinar si son las mismas o no. Los jugadores aquí, si intercambian sus respuestas.

Condición para ganar: Deben determinar si se les ha dado o no entradas iguales. TagATune aplica específicamente esta condición para la asignación de puntos, los mismos que se acreditan sólo si ambos jugadores han contestado correctamente.

Juegos basados en el mecanismo Inversion-Problem

Son el resultado generalizado de estos tres diferentes juegos: Peekaboom, Phetch y Verbosity:

Configuración inicial: Se escoge a 2 jugadores aleatoriamente entre los disponibles.

Reglas: A un jugador se le asigna el rol de “descriptor” y al otro como “adivinator”. Al descriptor se le da una entrada, y basada en ella debe producir respuestas que son enviadas al adivinator. Las respuestas del descriptor deben ayudar al compañero a determinar la respuesta original.

Condición para ganar: El adivinator produce la entrada que originalmente le fue dada al descriptor. En Verbosity la entrada es una palabra y las salidas son hechos de sentido común relacionados con esa respuesta. La idea básica es que a los jugadores no se les pregunte directamente hechos de la palabra dada, sino que el juego es diseñado para que estos hechos sean colectados como efecto colateral del juego. Para ello, se les dice a los jugadores, por ejemplo: “por favor ingrese factores acerca de la leche”.

“Inversion-Problem” se diferencia de los otros mecanismos, porque no mantiene a los jugadores con la misma tarea, después de cada ronda, sus roles pueden cambiar (así el adivinator se vuelve el descriptor y viceversa).

Tanto “Input-Agreement” como “Inversion-Problem” son mecanismos basados en la denominada “Función de Cómputo”, que propone que además del cómputo principal, se debe realizar una función adicional. En el caso específico de TagATune la Función de Cómputo es indicar si las entradas son iguales o diferentes [18].

2.2.2. Aplicaciones

Los GWAP tienen una amplia gama de aplicaciones en áreas tan diversas como Seguridad Web, Visión por Computadora, Accesibilidad en Internet, Filtrado de Contenido para adultos, y Búsquedas por Internet [8]. Algunos ejemplos de GWAP son presentados en la Tabla I.

Tabla I - Juegos de Human Computation

Juego	Descripción	Mecanismo	Problema de Inteligencia Artificial
ESP Game	Los jugadores deben coincidir en las etiquetas a ingresar para la misma imagen.	Output-Agreement	Reconocimiento de Objetos.
Peekaboom	El jugador 1 revela partes de la imagen	Función de Cómputo	Localización de Objetos

	asociadas a la palabra oculta y el jugador 2 debe adivinar esa palabra secreta.	(Inversion-Problem)	
Verbosity	El jugador 1 describe las propiedades de la entidad asociada a la palabra secreta y el jugador 2 debe adivinarla.	Función de Cómputo (Inversion-Problem)	Extracción de Conocimiento
TagATune	Los jugadores intercambian etiquetas y determinan si los 2 clips de sonido dados, son iguales o diferentes.	Función de Cómputo (Input-Agreement)	Clasificación de Música
HerdIt	Los jugadores escogen los tags o etiquetas que mejor describan una música. En "Categorilla", por ejemplo, los jugadores nombran entidades que se ajusten a una plantilla indicada.	Output-Agreement	Procesamiento de Lenguaje Natural
MoodSwings	Los jugadores dan click en una grilla 2D para indicar el balance y la intensidad del estado de ánimo que proporciona un clip	Output-Agreement	Clasificación de Estados de Ánimo de una música

	de música.		
Phrase Detective	Los jugadores identifican relaciones entre palabras y frases en una descripción breve.	Output-Agreement	Procesamiento de Lenguaje Natural
Phylo	Los jugadores alinean bloques coloreados moviéndolos horizontalmente e insertando espacios	Output-Agreement	Alineación de Genoma

Fuente: Edith Law y Luis von Ahn, Human Computation. Morgan & Claypool Synthesis Lectures on Artificial Intelligence and Machine Learning, Agosto 2011.

2.2.3. Calidad de las etiquetas

Los diseñadores de juegos de Human Computation se enfrentan con el desafío de construir un sistema que cumpla simultáneamente dos objetivos: satisfacer a los jugadores y llevar a cabo un procesamiento eficiente y preciso. Por un lado, se puede diseñar un juego que atraiga a muchos jugadores, pero que no recoja datos útiles, y por otro lado, si el juego está diseñado para recopilar los datos más limpios posibles, sin prestar atención a cuán divertida es la tarea, entonces ningún jugador estaría interesado en participar.

Para los juegos basados en mecanismos que incluyen Función de Cómputo, esta desventaja es evidente, puesto que dan prioridad a

mantener motivado al jugador que a obtener mejores resultados, y por lo tanto, un post-procesamiento relevante es requerido [18]. Lo mejor es diseñar mecanismos en que los jugadores sean naturalmente motivados. Herdagdelen y Baroni [19], indican que el diseño de la tarea también puede afectar directamente la calidad de la salida final.

Los incentivos además de ser una forma de atraer la participación de jugadores, son una posible solución para motivarlos a ingresar etiquetas específicas [18].

Otras formas de obtener buenos resultados son las observadas en juegos como el ESP y KissKissBan. En el ESP, se aplica por ejemplo, las denominadas “taboo-words” o palabras bloqueadas, como una forma de obligar al jugador a ser más creativo en el ingreso de etiquetas, impidiendo así, el ingreso de aquellas muy generales [4].

Algo parecido se implementa en KissKissBan [20], con la variante de que ya no son sólo parejas de jugadores, sino que se introduce un nuevo rol, un tercer jugador que actúa como el moderador de las etiquetas, impidiendo a los otros dos el ingresar aquellas que él indique.

Esto además hace que el juego sea competitivo y colaborativo a la vez, y que el control de palabras bloqueadas dependa del criterio de otro humano.

Estas dos soluciones son una forma de decirle explícitamente al usuario, que esas son las palabras que no debe ingresar. Tras observar estas ventajas del mecanismo “Output-Agreement”, cabe indicar que los mecanismos a pesar de ser generalizaciones, no son aplicables a todos los problemas. Por ejemplo, el mecanismo “Output-Agreement” se basa en el hecho de que es fácil conseguir coincidencias de etiquetas para una imagen [18], pero no así para videos, música o sonidos, como es el caso en el que mejor se ajustan los juegos basados en “Input-Agreement”.

2.2.4. Estrategias para evitar acuerdos entre jugadores

Herdagdelen y Baroni [19] indican que una forma de evitar que los jugadores hagan trampas es introduciendo hechos cuyas respuestas ya sean conocidas por el sistema, con el fin de determinar si un jugador está verdaderamente participando en el juego. Sin embargo, esto requiere una gran colección de hechos comprobados que no son fáciles

de obtener. Por ello lo mejor es diseñar mecanismos en que los jugadores sean naturalmente motivados y jueguen sin malas intenciones[18].

Von Ahn y Dabbish [4] presentan varios mecanismos para contrarrestar las trampas en el juego ESP y son los siguientes:

- El servidor del juego inicia una nueva partida cada 30 segundos: cuando un nuevo jugador se autentica, debe esperar hasta el siguiente límite de 30 segundos para emparejarlo con otro jugador y empezar su juego. Esto se hace para asegurar que los jugadores se emparejen al azar y no logren hacer trampas autenticándose al mismo tiempo que sus compañeros.
- Como los jugadores forman parejas aleatorias, la probabilidad de que estos se pongan de acuerdo en un estrategia autenticándose al mismo tiempo es baja, porque no tendrán información sobre quién es su pareja.
- Midiendo el tiempo promedio en que los jugadores ingresaron sus etiquetas puede reflejar si se ha intentado hacer trampas estratégicamente entre varios jugadores, pues una marcada

disminución del tiempo promedio debe indicar un acuerdo masivo en una estrategia.

2.2.5. Incentivos

Otro factor importante en los sistemas de Human Computation son los incentivos, los cuales afectan el comportamiento de los participantes incluyendo: si realizan la tarea en su totalidad, o nivel de participación, la tarea que escogen y lo bien que la realizan [18].

Los incentivos pueden ser de 2 tipos, según la forma de recompensar al participante:

Aquellos provocados por **Motivaciones intrínsecas**: sea por altruismo (por el beneficio de otros), el deseo de aprender o por competencia.

Motivaciones extrínsecas: como las recompensas monetarias (virtuales o reales) o por puntos y otras mecánicas de juegos.

“Cuánto incentivo proveer” y “Encontrar formas de asegurar que el sistema de incentivos sea resistente a la manipulación”, son temas que se tratan en mayor profundidad en áreas fuera de Human Computation. El tema de cuánto incentivo proveer implica un entendimiento de la

psicología de los participantes. Sin embargo, existen evidencias de las que se concluye que, incentivos como los monetarios deben ser muy bien diseñados, para que puedan mejorar el rendimiento del modelo sin perjudicarlo a largo plazo; de no ser así, mejor tratar de excluirlos [21]. El diseñar sistemas de incentivos resistentes a la manipulación es relevante y específico de áreas como Teoría de Juegos y Diseño de Mecanismos[18].

CAPÍTULO 3

ANÁLISIS Y DISEÑO DE LA APLICACIÓN

3.1. Análisis de la solución

El juego 'CazaVideos' se diseñará de tal manera que varios jugadores puedan producir metadatos de forma conjunta. Dado que los usuarios accederán a la aplicación desde diversas plataformas, y que el contenido a etiquetar estará disponible "en línea", es importante que el juego sea diseñado como una aplicación Web. La solución también requiere utilizar contribuciones de grandes cantidades de usuarios, por lo que se aprovecharán los conceptos asociados a GWAP, específicamente el mecanismo "Input-Agreement" por ser éste ideal para obtener información sobre recursos de alto contenido descriptivo como videos, sonidos y música.

Una parte central en el diseño de la aplicación es la que se refiere a la interfaz. Las interfaces Web se actualizan localmente como respuesta a un evento del usuario que la maneja, y no responden a eventos de usuario en un cliente distinto. Debido a esto se debe diseñar un mecanismo en el que ambas interfaces puedan comunicarse y responder sincrónicamente a los eventos generados en cualquiera de los clientes que interactúan. Estas consideraciones de diseño se presentarán en este capítulo.

3.1.1. Requisitos funcionales

El diseño de la aplicación se regirá por los requisitos funcionales que se han identificado para la misma. A continuación una enumeración de ellos:

- La autenticación de un usuario a través de Facebook.
- Crear una nueva partida.
- Asignar automáticamente compañeros de juego, es decir, dos jugadores por sesión. (Donde, el máximo está determinado por el número de sesiones soportadas en el servidor).

- Asignar un conjunto de fragmentos de video para las rondas de la partida a iniciar.
- Reproducir un fragmento de vídeo de YouTube para ambos jugadores de manera sincrónica. Dicho video puede ser aleatoriamente igual o diferente entre compañeros, pero debe pertenecer a una misma categoría.
- Mostrar una sección para que los jugadores ingresen palabras o frases describiendo el fragmento de vídeo actual, siempre que el jugador cuente con tiempo disponible en la partida.
- Visualizar las palabras o frases descritas por su compañero de juego en tiempo real.
- Mostrar un control para permitir al jugador decidir si el vídeo es igual o diferente al de su compañero.
- Acumular automáticamente puntos cada vez que ambos compañeros están de acuerdo en la decisión sobre el video.
- Mostrar resultados del juego, puntajes y resumen del mismo al finalizar una partida.
- Visualizar las bonificaciones de puntos al llegar a un determinado nivel en los puntajes acumulados.

- Terminar la sesión de juego en el momento en el que los jugadores lo deseen.

3.1.2. Requisitos no funcionales

Existen requisitos no funcionales asociados a la importancia de mantener el interés del jugador en la aplicación. Los requisitos no funcionales identificados son:

- Diseñar una interfaz usable, sencilla con un diseño elegante y claro.
- Dar constantemente retroalimentación al usuario.
- Proporcionar un ambiente desafiante y entretenido.
- Lograr compatibilidad con los principales navegadores usados en el entorno ecuatoriano, estos son los de Microsoft, Google y Mozilla.

3.2. Diseño de la solución

La aplicación a desarrollar será implementada utilizando un lenguaje de programación Web que permita generar páginas de forma dinámica. El control de las acciones del juego será implementado en diversos módulos

según la funcionalidad implementada. A continuación se enumeran dichos módulos y se detalla el objetivo de cada uno de ellos:

- Elaboración de la Mesa.
- Control de la Mesa.
- Sincronización de interfaces.

Elaboración de la Mesa: Módulo en el que se incluye la lógica a implementar para iniciar una sesión de juego. Esto incluye tanto la autenticación como las rutinas para asociar en parejas los usuarios disponibles.

Control de la Mesa: Módulo en el que se implementarán las rutinas que controlan los eventos que se disparan durante el desarrollo del juego. Entre las rutinas a implementar están las validaciones para decidir si el juego continúa o no, el manejo de tiempo, acumulación de puntos y cambios de estado.

Sincronización de interfaces: En este módulo se implementará el backend que permite la sincronización de mensajes y decisiones entre las parejas

de jugadores. Para la implementación se utilizará el método de comunicación conocido como “WebSockets” [22], determinado en el siguiente análisis.

Tabla II – Tabla Comparativa de Polling, Long Polling y WebSockets

HTTP Polling	HTTP Long Polling (Ajax Push, Técnica Comet)	WebSockets
<p>Hace consultas constantes, realiza muchos requerimientos HTTP, solicita cambios al servidor cada cierto intervalo de tiempo (usando AJAX y procesando la respuesta en el servidor).</p> <p>Conexión: Establece una conexión unidireccional.</p> <p>Consumo de ancho de banda: requiere conexión para cada mensaje y para aquellos mensajes sin información.</p> <p>Hay un alto consumo de recursos y ancho de banda.</p> <p>Si se reduce el intervalo de tiempo de atención de los requerimientos, aumentará el número de los mismos y se obtendrá incluso muchos mensajes sin respuestas.</p>	<p>Es una variación de la técnica de polling tradicional, el servidor intenta "mantenerse abierto" y no responder de inmediato a cada solicitud HTTP, sino hasta cuando ocurra un evento y haya información disponible para el cliente, esto es, para evitar enviar una respuesta HTTP vacía.</p> <p>Una vez que la información está disponible (o después de un tiempo de espera adecuado), una respuesta completa se envía al cliente. De ahí que el cliente normalmente, de inmediato vuelve a hacer el requerimiento.</p> <p>Conexión: Establece una conexión unidireccional.</p> <p>Consumo de ancho de banda: requiere conexión para cada mensaje.</p>	<p>No requiere de complementos o plugins, como en el caso de Flash o Silverlight.</p> <p>Es construido nativamente en los navegadores.</p> <p>Conexión: Establece una conexión bidireccional. La conexión se mantiene viva. Puede escribir y recibir datos en JavaScript como si estuviera usando un socket TCP puro.</p> <p>Consumo de ancho de banda: requiere conexión para realizar el saludo o handshake de configuración inicial y unos pocos bytes por trama.</p> <p>Requiere un sencillo uso de JavaScript, y el api está siendo estandarizado por la W3C.</p>

<p>Aumento de latencia²</p> <p>Complejidad en JavaScript</p> <p>Podría sobrecargarse el servidor y dejar de prestar servicios, al no poder abastecer la cantidad de solicitudes.</p>	<p>Una desventaja es el uso de una cabecera de datos adicional, que se necesita enviar de vuelta en la respuesta, lo que causa una sobrecarga adicional.</p> <p>Al mantener una conexión abierta, disminuye la latencia.</p> <p>Usa menos ancho de banda que Polling.</p>	<p>No se requiere de mayor esfuerzo para entender y analizar la información recibida.</p> <p>El Protocolo WebSocket está siendo normalizado por la IETF.</p> <p>La ventaja sobre Long Polling es la eliminación de cabeceras extras, en la apertura y cierre de conexión del socket para atender un requerimiento.</p>
---	---	--

3.2.1. Diseño Lógico de la aplicación

El Diseño Lógico de la aplicación encapsula la relación que tienen las diversas tecnologías usadas para implementar los módulos descritos.

² Latencia: Suma de retardos temporales dentro de una red, producido por la demora en la propagación y transmisión de paquetes dentro de la red.

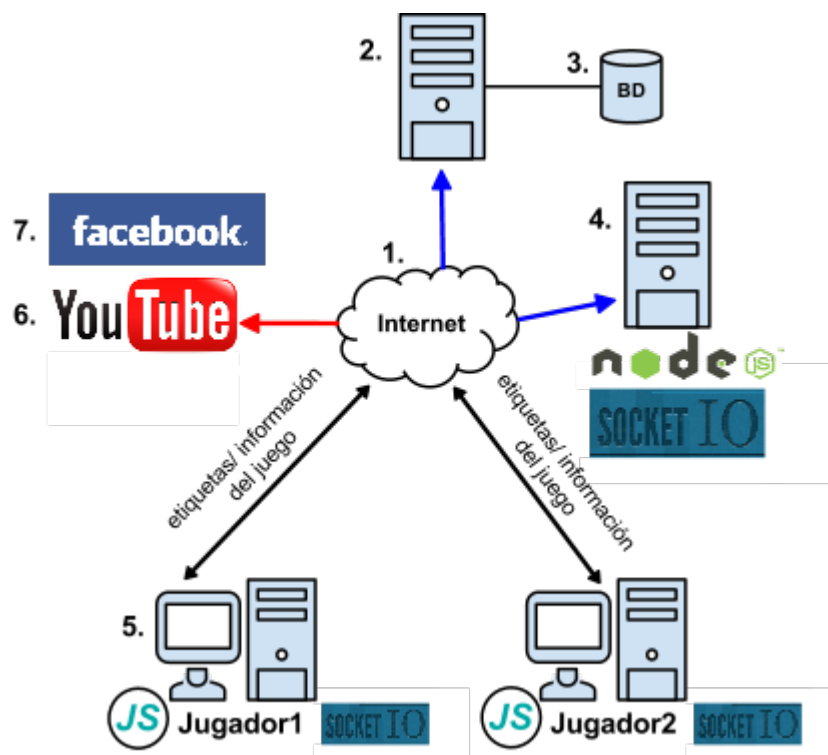


Figura 3. 1 - Diseño lógico de 'CazaVideos'

1. Internet.
2. Servidor de Aplicaciones con la aplicación web.
3. Base de datos.
4. Servidor de sockets.
5. Cliente.
6. Repositorio de videos YouTube.
7. Facebook para obtener datos del usuario autenticado.

En la Figura 3.1 se puede observar el flujo de información requerido para el desarrollo del juego. Los clientes entre sí y con el servidor intercambian: mensaje o etiqueta, tiempo de juego, vídeo y decisión. Otro factor importante a notar es que la aplicación no almacena los vídeos, estos son tomados directamente de los servidores de YouTube y presentados en la interfaz del juego.

En la Figura 3.2, se representa la aplicación del concepto “Input Agreement” como una solución al sistema, en el que *las entradas* representan a los fragmentos de vídeo. El *proceso* es la acción de los jugadores de producir etiquetas asignadas a sus respectivos fragmentos de vídeo y las *salidas* son las etiquetas que se generaron a medida que transcurría el tiempo del juego.

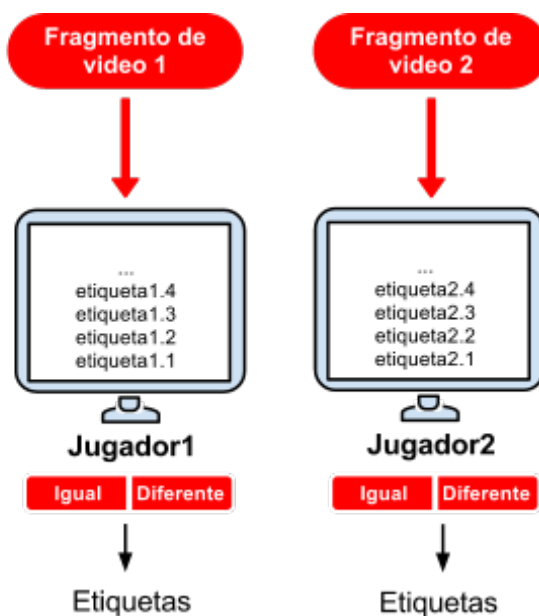


Figura 3. 2 - Mecanismo de "Input-Agreement" aplicado en 'CazaVideos'

3.2.2. Interfaz de usuario

La interfaz principal del juego está compuesta por tres secciones. La primera corresponde a la parte central de la interfaz, que contiene el video sin controles de reproducción, con la finalidad de evitar que el jugador se salte a otro intervalo que no sea el asignado. La segunda es la parte lateral, que incluye los contenedores, como se puede apreciar en la Figura 3.3 En los contenedores se visualizan las etiquetas ingresadas por el jugador actual (izquierda) y su compañero (derecha). Finalmente, se ha colocado en la parte superior derecha el reloj de conteo y en la parte superior izquierda el puntaje adquirido.

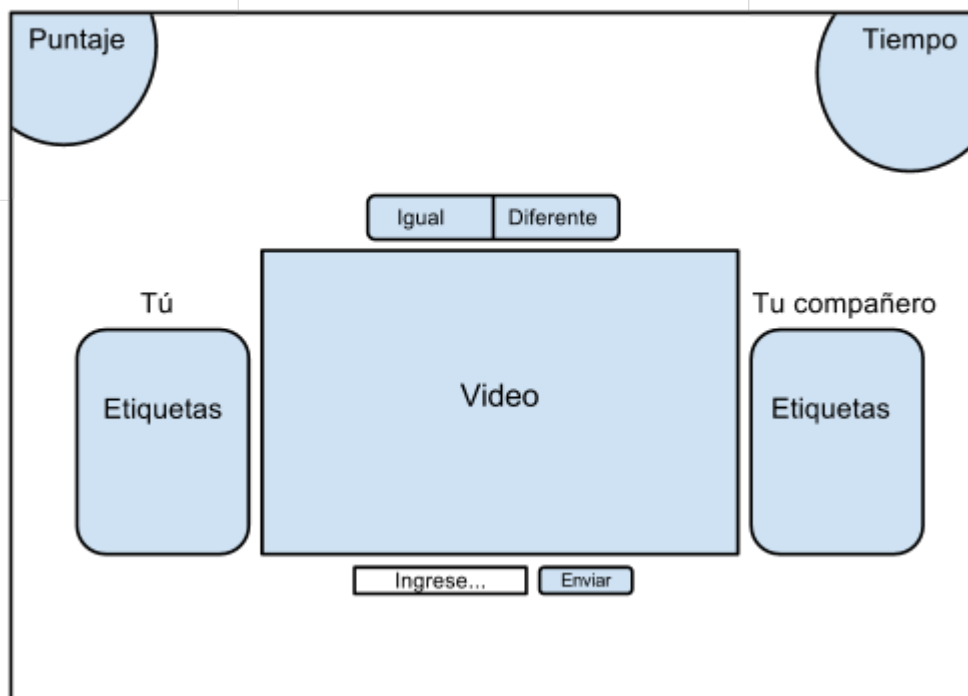


Figura 3. 3 - Interfaz principal del juego

Las interfaces de fin de juego, correspondientes al resumen del juego y la de ranking (puntajes) se reparten como se ilustra en la Figura 3.4:

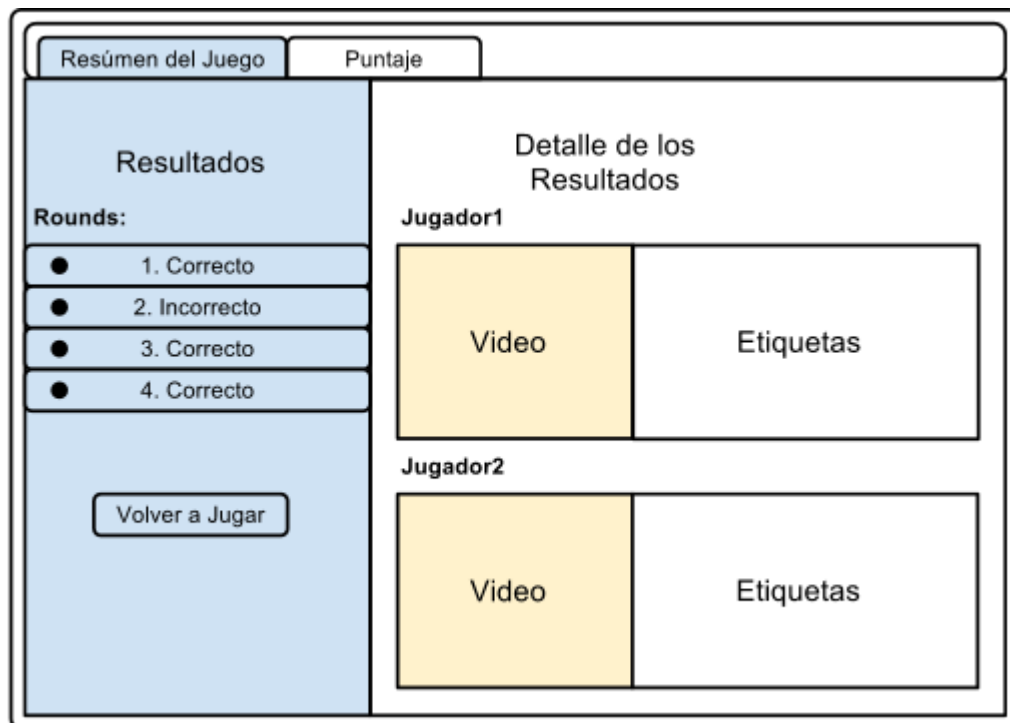


Figura 3. 4 - Interfaz de Game Recap

El Resumen del Juego, o Game Recap, incluye dos áreas principales: Resultados y Detalle de los resultados. El área de resultados incluye botones en forma de pestañas, con etiquetas que muestran las calificaciones grupales, obtenidas en la participación colaborativa de los jugadores de cada una de las rondas de la mesa.

El área de Detalle de los Resultados muestra el vídeo y las etiquetas que escribieron los jugadores, en forma de resumen de la ronda seleccionada.

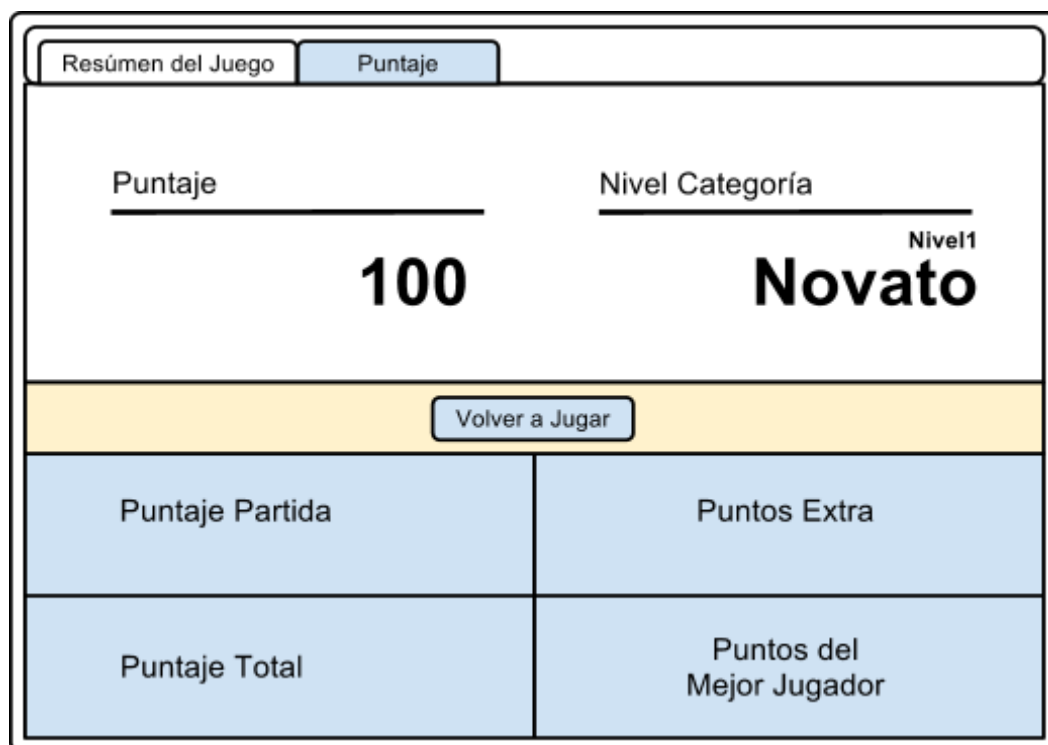


Figura 3. 5 - Interfaz de Ranking o Puntajes

Como se ve en la Figura 3.5, la Interfaz de Ranking o Puntajes incluye el puntaje de la partida actual y otros que le resulten de interés como: *Puntaje Total*, acumulado en todas sus partidas; *Puntos Extra*, que obtuvo como bonificación ya sea por contestar más rápido o llegar al

máximo de puntos adquirido de un nivel; y Puntos del *Mejor Jugador*, como motivación para alcanzar esa meta.

3.2.3. Filtrado de la información

El ingreso de etiquetas no se restringe a un vocabulario predefinido. Se permitirá el ingreso de todas las palabras escogidas por el jugador y se las filtrará al final realizando un post-procesamiento para obtener las etiquetas oficiales. Cabe indicar que son muy pocas las etiquetas que se usan para hacer trampa, pues estas normalmente resultan de ponerse de acuerdo con el compañero usando el intercambio de mensajes como medio de comunicación [18]. El hecho de que las parejas de jugadores sean escogidas aleatoriamente y que cada fragmento de vídeo sea reproducido por únicamente 30 segundos, son también una forma de garantizar que los usuarios no podrán comunicarse entre sí. Al final se presentará a los usuarios un resumen del puntaje de su participación durante la sesión de juego, con la finalidad de que puedan aprender de sus errores.

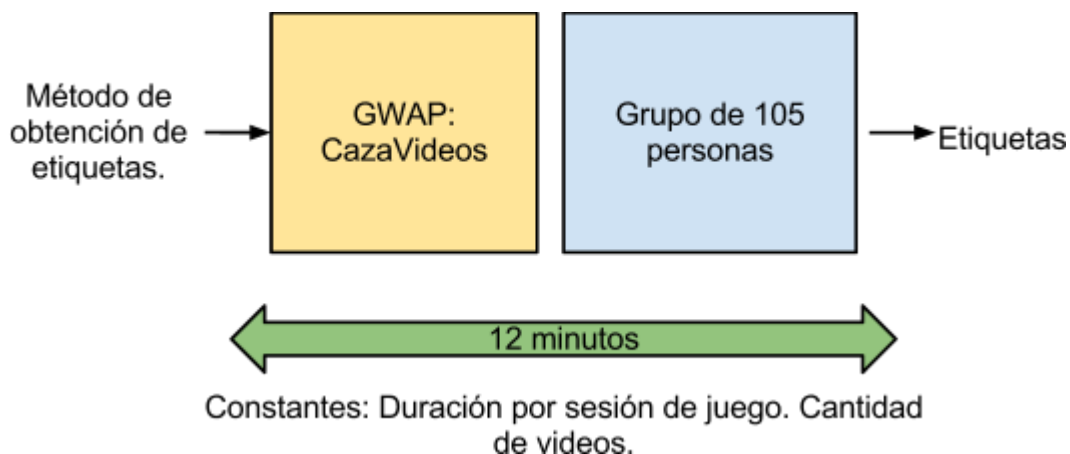
Se implementará una rutina que agrupe las etiquetas en función del fragmento de vídeo, luego a cada grupo se aplican algoritmos para

excluir anotaciones que no sean relevantes. Para efectos del alcance de este trabajo, los datos que se filtrarán serán: caracteres especiales y palabras de apoyo a la oración, conocidas como *Stop Words* o *Palabras Vacías*, como por ejemplo los artículos, preposiciones, etc. Adicionalmente se utilizará el proceso de *stemming*, el cual permite detectar etiquetas cuya intención sea describir la misma palabra, como por ejemplo las conjugaciones de verbos.

Cabe decir que la gran cantidad de datos redundantes facilita encontrar anotaciones válidas entre aquellas que tienen frecuencia más alta. Las etiquetas finales serán presentadas como *nubes de tags* donde se verán resaltadas las palabras con mayor frecuencia.

3.2.4. Diseño de plan de pruebas

La presente investigación se llevará a cabo usando un diseño cuasi-experimental ya que la muestra fue escogida de forma no aleatoria, conformada por estudiantes de ingeniería en su mayor parte. En la Figura 3.6 se puede observar el tiempo que cada usuario tendrá para generar las etiquetas, el cual corresponde a dos sesiones de juego, o 12 minutos de duración.



Variable Independiente: Método de Obtención de etiquetas.

Variable Dependiente: Calidad, que se refiere a la eficiencia del método de recolección de etiqueta.

Figura 3. 6 - Diseño cuasi-experimental.

Se prevé que los usuarios trabajen 16 horas, el mismo número de horas requerido que una persona se demoraría en etiquetar únicamente los fragmentos educativos, a las que, para prevenir errores con un 76% de confianza³ aproximando a 21 horas de trabajo, se les agrega 5 horas adicionales. Por lo tanto se escogerán 105 personas, cada una jugará las dos sesiones de juego establecidas, para probar la capacidad de generación masiva de etiquetas

³ 5h adicionales dentro de las 21h totales $5/21=0.24$ $1-0.24 = 0.76$

Tipos de videos. Los fragmentos de vídeo pueden ser de tres tipos: *Educativos, Blooper* y *Popfail*. En la categoría *Educativos*, los vídeos que se extraerán serán obtenidos del canal de YouTube de ESPOL, los cuales presentan sesiones de clase de las áreas: física, matemáticas o química. Los de tipo *Blooper*, presentan accidentes casuales y los de tipo *Popfail*, presentan situaciones inesperadas durante interpretaciones de artistas populares. Las anotaciones generadas en el experimento serán agrupadas según su fragmento de vídeo y su frecuencia, y de estas se medirá visualmente la calidad basado en su relevancia con el tema y su generalidad.

Para comprobar la correcta funcionalidad de la aplicación, se realizarán pruebas enfocadas tanto en el funcionamiento general del juego, como en los módulos de sincronización.

Con respecto a las pruebas de funcionamiento general del juego, se revisarán las acciones importantes que conlleva una sesión de juego, como son:

- Prueba de inicio de una partida.
- Anotación de un video dentro del tiempo establecido.
- Consulta de Resumen de Juego.

Prueba de inicio de una partida: Al dar click en "Iniciar", deberá aparecer la Interfaz del Juego enmascarada con un fondo gris y transparencia, mientras se sincroniza el video de los jugadores, para la primera ronda.

Anotación de un video dentro del tiempo establecido: Al iniciarse una nueva ronda y al haber esperado la sincronización de su respectivo video, se revelará la interfaz del juego y empezará el conteo regresivo de los 30 segundos establecidos. De tal manera que el jugador pueda ingresar anotaciones sobre lo observado en el video durante ese lapso. Una vez finalizado el conteo, nuevamente se cubrirá la interfaz para cargar la ronda siguiente, impidiendo al jugador el ingreso de más anotaciones.

Consulta de Resumen de Juego: Esta se comprobará al final de las 10 rondas disponibles o cronómetro global del juego, momento en el que aparecerá la ventana de finalización, que incluye tanto la sección de

Puntajes como la de *Resumen de Juego*, la misma que representa la recopilación de todas las rondas participadas, el resultado obtenido por ronda, sea: *CORRECTO*, *INCORRECTO* o *NO_CONTESTO*, y su respectivo detalle: el video que le fue asignado y las etiquetas que ingresó contra las de su compañero.

Con el fin de probar los módulos que implican la sincronización o acciones simultáneas, se considerarán tres características, que se muestran en la Tabla III, dada su importancia para la obtención de las anotaciones.

Número de prueba	Característica
1	Sincronización de fragmentos de vídeo
2	Sincronización de mensajes
3	Acción de decisión "Igual o diferente"

Tabla III - Características incluidas en las pruebas de funcionalidad

Las pruebas funcionales serán realizadas por los autores de este documento, utilizando dos ordenadores de similares características y la misma velocidad de internet.

Métricas

En el experimento se busca comprobar que los usuarios emparejados, a los que llamaremos A y B, estén sincronizados, por lo tanto las características antes mencionadas serán ejecutadas tres veces y se anotará la marca de tiempo en la que estas sucedan para cada usuario. Las unidades serán almacenadas utilizando el huso horario de Guayaquil, Ecuador: GMT-5.

Número de prueba	Métrica	Propósito	Información requerida
1	timevid_jA	Comparar el momento de reproducción del fragmento con el del jugador B	Timestamp en que empieza a reproducir el vídeo del jugador A
1	timevid_jB	Comparar el momento de reproducción del fragmento con el del jugador A	Timestamp en que empieza a reproducir el vídeo del jugador B
2	timech_sjA	Comparar el momento de envío del chat del jugador A con el de recepción del chat del jugador B	Timestamp en que el jugador A envía un mensaje de chat
2	timech_rjB	Comparar el momento de recepción del chat del jugador B con el de envío del chat del jugador A	Timestamp en que el jugador B visualiza un mensaje de chat

3	timede_sjA	Comparar el momento de envío de decisión del jugador A con el de recepción de decisión del jugador B	Timestamp en que el jugador A envía la señal de decisión
3	timede_rjA	Comparar el momento de decisión del jugador B con el de decisión del jugador A	Timestamp en que el jugador B visualiza la señal de decisión enviada por el jugador A

Tabla IV - Métricas incluidas en las pruebas de funcionalidad

Casos de Prueba

1. Sincronización de fragmentos de vídeo

Caso de prueba	Detalle
Número y nombre del caso	1. Sincronización de fragmentos de vídeo
Criterios de inicio	Ambos jugadores (A y B) deben estar autenticados vía Facebook en la aplicación 'CazaVídeos'
Procedimiento	<ul style="list-style-type: none"> • Jugadores dan click en iniciar juego nuevo. • Jugadores observan la pantalla principal del juego. • Jugadores reciben la notificación de sincronización de vídeo. • Anotar el timestamp en que los vídeos empiezan su reproducción para ambos jugadores (A y B).
Resultado esperado	El timestamp de las anotaciones de ambos comienzos de reproducción debe ser igual o muy cercana.

Tabla V - Caso de Prueba 1 – Sincronización de fragmentos de video

2. Sincronización de mensajes

Caso de prueba	Detalle
Número y nombre del caso	2. Sincronización de mensajes
Criterios de inicio	Ambos jugadores (A y B) deben estar autenticados vía Facebook en la aplicación 'CazaVídeos'
Procedimiento	<ul style="list-style-type: none"> • Jugadores dan click en iniciar juego nuevo. • Jugadores observan la pantalla principal del juego. • Jugadores reciben la notificación de sincronización y se empieza a reproducir el vídeo. • Jugador A envía un mensaje de chat. • Se anota inmediatamente el timestamp de envío de mensaje. • Jugador B recibe un mensaje de chat. • Se anota inmediatamente el timestamp de recepción del mensaje.
Resultado esperado	El timestamp de las anotaciones de envío y recepción de mensajes debe ser igual o muy cercana.

Tabla VI - Caso de Prueba 2 – Sincronización de mensajes

3. Acción de decisión “Igual o diferente”

Caso de prueba	Detalle
Número y nombre del caso	3. Acción de decisión “Igual o diferente”
Criterios de	Ambos jugadores (A y B) deben estar autenticados vía

inicio	Facebook en la aplicación 'CazaVÍdeos'
Procedimiento	<ul style="list-style-type: none"> • Jugadores dan click en iniciar juego nuevo. • Jugadores observan la pantalla principal del juego. • Jugadores reciben la notificación de sincronización y se empieza a reproducir el vídeo. • Jugador A decide contestar "Igual". • Se anota inmediatamente el timestamp de envío de decisión. • Jugador B recibe notificación de que su compañero ha tomado una decisión. • Se anota inmediatamente el timestamp de recepción de decisión.
Resultado esperado	El timestamp de las anotaciones de envío y recepción de decisión debe ser igual o muy cercana.

Tabla VII - Caso de Prueba 3 – Acción de decisión "Igual o diferente"

En el capítulo siguiente se mostrarán los detalles de implementación de la aplicación siguiendo los parámetros de diseño arriba propuestos.

CAPÍTULO 4

IMPLEMENTACIÓN Y PRUEBAS

4.1. Plataforma tecnológica

Para la implementación de CazaVideos, además del servidor de aplicaciones se utilizarán dos plataformas externas: Youtube y Facebook. Dichas plataformas poseen Interfaces de Programación de Aplicaciones (API por sus siglas en inglés) sencillas, completas y de uso gratuito.

También se analizaron herramientas que permiten implementar aplicaciones en tiempo real, para realizar la actualización constante del estado de las partidas y permitir las sincronizaciones requeridas. Las secciones siguientes presentan detalles sobre estas herramientas.

4.1.1 YouTube

Es un sitio Web que brinda un servicio gratuito para compartir videos. Es una plataforma administrada por Google y actualmente el mayor

repositorio de videos del mundo. Cuenta con un API sencillo en JavaScript para la publicación de videos desde otras páginas Web.

Se escogió a YouTube como base para presentación de videos en la aplicación, según un listado previamente seleccionado para cada categoría mencionada en el capítulo anterior.

4.1.2. Facebook

Es un sitio Web que ofrece acceso a la red social más grande del mundo. Es una plataforma abierta a cualquier persona con una cuenta de correo, y ha captado la atención de muchos usuarios, desarrolladores y personas de negocio.

La librería del cliente oficial de Facebook está disponible en el lenguaje de programación PHP. Usa el protocolo de autenticación y autorización OAuth 2.0, que es un protocolo abierto para permitir una autenticación segura del API de forma simple y estandarizada tanto para aplicaciones Web como de escritorio.

Se escogió a Facebook como plataforma social para la creación de la aplicación y su rápida difusión en la red.

4.1.3. Node.js y Socket.io (WebSockets)

Luego de analizar las ventajas y desventajas de tecnologías para la implementación de aplicaciones sincrónicas en tiempo real como Comet, Reverse Ajax, Long Polling, entre otras, se escogió Websockets. Este último, es un protocolo que permite un mecanismo eficaz de comunicación al proveer un canal bidireccional entre cliente y servidor para enviar mensajes, evitando tráfico innecesario. Esto ayuda a una comunicación más rápida y un aprovechamiento más efectivo del ancho de banda.

Éste estándar simplifica mucho la complejidad de la comunicación Web bidireccional y el manejo de las conexiones. El protocolo para el lado servidor está siendo normalizado por el Grupo Especial sobre Ingeniería de Internet o IETF, por sus siglas en inglés, y el API de Websockets para el lado cliente está siendo estandarizado por la W3C.

El protocolo realiza el *handshake* inicial y el entramado básico del mensaje sobre TCP, para proporcionar un mecanismo a las aplicaciones basadas en navegadores que necesiten una comunicación en dos vías con servidores, evitando múltiples conexiones HTTP (usando XMLHttpRequest o Iframes y Long Polling).

Se buscaron herramientas, marcos de trabajo que permitan simplificar aún más la complejidad de los Websockets y APIs que permitan una rápida implementación del servidor, escogiendo así a Node.js por proveer un API sencillo, amable con JavaScript y con varias ventajas al combinarlo con Socket.io, entre las que están: Poder contar con el mismo lenguaje tanto en el cliente como en el servidor y poder crear una aplicación cross-browser, es decir, que pueda ser soportada por la mayoría de los navegadores o los más usados.

Node.js es un entorno de ejecución y una librería, que usa un modelo basado en eventos, entradas y salidas no bloqueantes y por lo tanto asíncrona, ideal para uso intensivo de datos en aplicaciones de tiempo real que se ejecutan a través de dispositivos distribuidos.

Node.js permite ejecutar código JavaScript en el servidor, para la implementación de aplicaciones en las que se requiere una actualización de datos continua, sin llamadas constantes. Para ejecutar el código JavaScript en el lado del servidor Node.js lo interpreta y ejecuta haciendo uso de la Máquina Virtual V8 de Google, el mismo entorno de ejecución que Google Chrome utiliza.

Node.js provee muchos módulos útiles, entre ellos Socket.io, que es un módulo de Node.js soportado por navegadores como: Internet Explorer, Safari, Google Chrome, Firefox y Opera, y que permite una comunicación en tiempo real entre un servidor Node.js y sus clientes, mediante la implementación de WebSockets.

4.2. Arquitectura de la aplicación

Se separó la lógica en dos partes que, individualmente, permitan una comunicación asíncrona entre cliente y servidor (ver Figura 4.1). La primera corresponde a la implementación del servidor de sockets realizada en Node.js incluyendo el módulo Socket.io, y la segunda, al cliente en JavaScript con socket.io.js (librería para acceder a Socket.io desde el cliente).

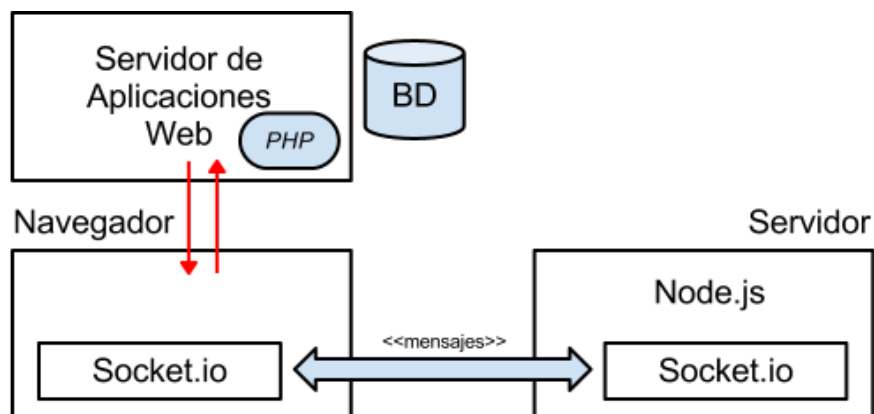


Figura 4. 1 – Arquitectura de la aplicación.

Se implementó la aplicación Web usando PHP con el Framework Symfony para guardar respaldo de los datos necesarios de la aplicación. Y se incluyó en ésta, al cliente de Socket.io.

La parte correspondiente al servidor de sockets procesa los mensajes que envía el cliente JavaScript, para que según sea el caso, devolver los mensajes correspondientes. Éste se usó para la implementación de las sincronizaciones existentes en el juego como son: El intercambio de etiquetas, la sincronización de videos y la sincronización de las respuestas “Igual o diferente”.

El cliente envía y recibe mensajes del servidor para comunicarse con otro. Tomamos como cliente a la aplicación implementada en Symfony

con la interfaz del juego, la misma que deberá intercambiar mensajes con el servidor basado en Node.js (ver Figura 4.2).

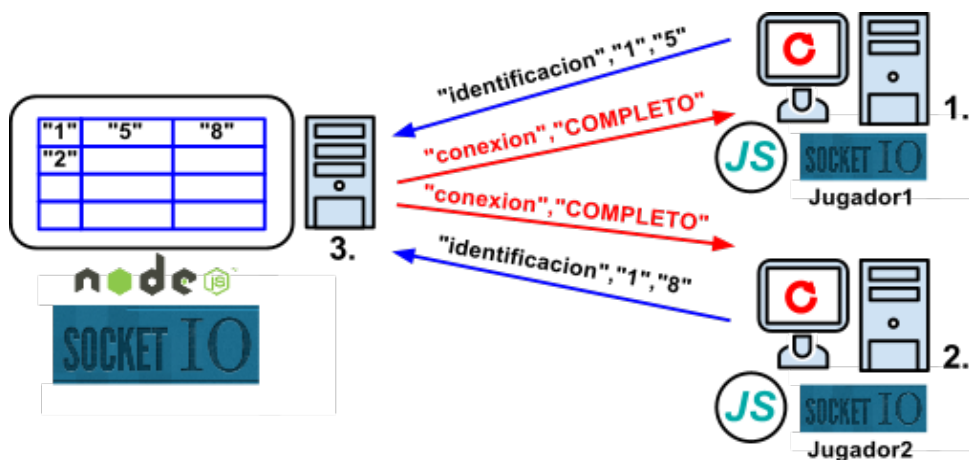


Figura 4. 2 - Intercambio de mensajes.

Tanto la identificación de los jugadores de una mesa, como la sincronización de videos, la sincronización de las respuestas "Igual o Diferente" y el intercambio de etiquetas mediante el chat, se realizaron de esta manera. Donde lo que cambia es la estructura donde se almacenan temporalmente los datos necesarios que envía el cliente, el tipo de mensaje y el contenido de éste, y la respuesta que les devuelve el servidor, codificados en formato JSON.

En el caso del chat, el mensaje enviado por un cliente no es entregado nuevamente a éste, sino sólo a su compañero de juego, el mismo que se

obtiene de una estructura a la que llamamos “mesas” creada en el servidor, y que registra los datos del usuario y su socket correspondiente al momento de identificarse.

4.3. Interfaz de usuario

Al ingresar a la aplicación, aparece el botón Entrar o Login de Facebook, para iniciar sesión en caso de no haberlo hecho. Una que ha entrado se observa el Menú Principal del Juego (ver Figura 4.3), que contiene sólo 2 opciones: “Iniciar” e “Instrucciones” (ver Figura 4.4).



Figura 4. 3 - Menú Principal del Juego.



Figura 4. 4 - Instrucciones del Juego.

Al dar click en “Iniciar”, se observa la interfaz del juego (ver Figura 4.5). En el diseño de esta interfaz, se utilizó la metáfora de un cine en casa, que tiene en el centro una pantalla y de fondo un gradiente color lila. En la parte superior con una mayor fuente se resaltó el cronómetro para que el usuario esté pendiente en cada momento del tiempo que dispone para continuar jugando.



Figura 4. 5 - Interfaz del Juego.

4.4. Calidad de etiquetas

Para garantizar etiquetas de calidad se ejecutó el diseño de filtrado de información definidos en la unidad 3.2.3, para lo que se desarrolló una rutina programada en el lenguaje Python, utilizando la librería de procesamiento de lenguaje natural NLTK, que permite realizar, entre otras cosas, filtrado de Stop Words y Stemming:

```
spanish_stops=set(stopwords.words("spanish"))
```

```
stemmer=SnowballStemmer("spanish")
```

Muchas librerías de procesamiento de texto requieren caracteres ASCII y los jugadores utilizan muchas veces caracteres latinos. De ahí que para utilizar estas rutinas, se agregó un encabezado especial que permite al compilador entender estos caracteres:

```
# -*- coding: iso-8859-15 -*-
```

4.5. Pruebas

Se realizaron pruebas a los datos recogidos durante el tiempo en que estuvo la aplicación en producción recaudando un total de 5,529 etiquetas en un total de 210 sesiones de juego.

Las 105 personas escogidas fueron expuestas al uso de la aplicación. La edad media de esta muestra es 23 años, con nivel de educación superior, 67% de sexo masculino, 33% femenino y estado civil soltero en su totalidad.

4.5.1. Pruebas de funcionalidad

Todas las pruebas propuestas en el capítulo 3.2.4 fueron realizadas en ordenadores con procesadores intel Core 2 duo, 2GB de memoria RAM, Sistema Operativo Windows 7, navegador *Google Chrome* y velocidad de Internet de 512 Kbps.

Prueba de inicio de una partida: Luego de dar click en “Iniciar”, efectivamente aparece una máscara indicando que se está esperando a un compañero de juego (ver Figura 4.6).



Figura 4. 6 - Inicio de Nueva Partida.

Anotación de un video dentro del tiempo establecido: La anotación dentro de los 30 segundos establecidos también fue exitosa, las etiquetas fueron ingresadas correctamente (ver Figura 4.7).



Figura 4. 7 - Obtención de Anotaciones durante el Juego.

Consulta de Resumen de Juego: Se logró satisfactoriamente visualizar los resultados obtenidos de la sesión de juego (ver Figura 4.8).

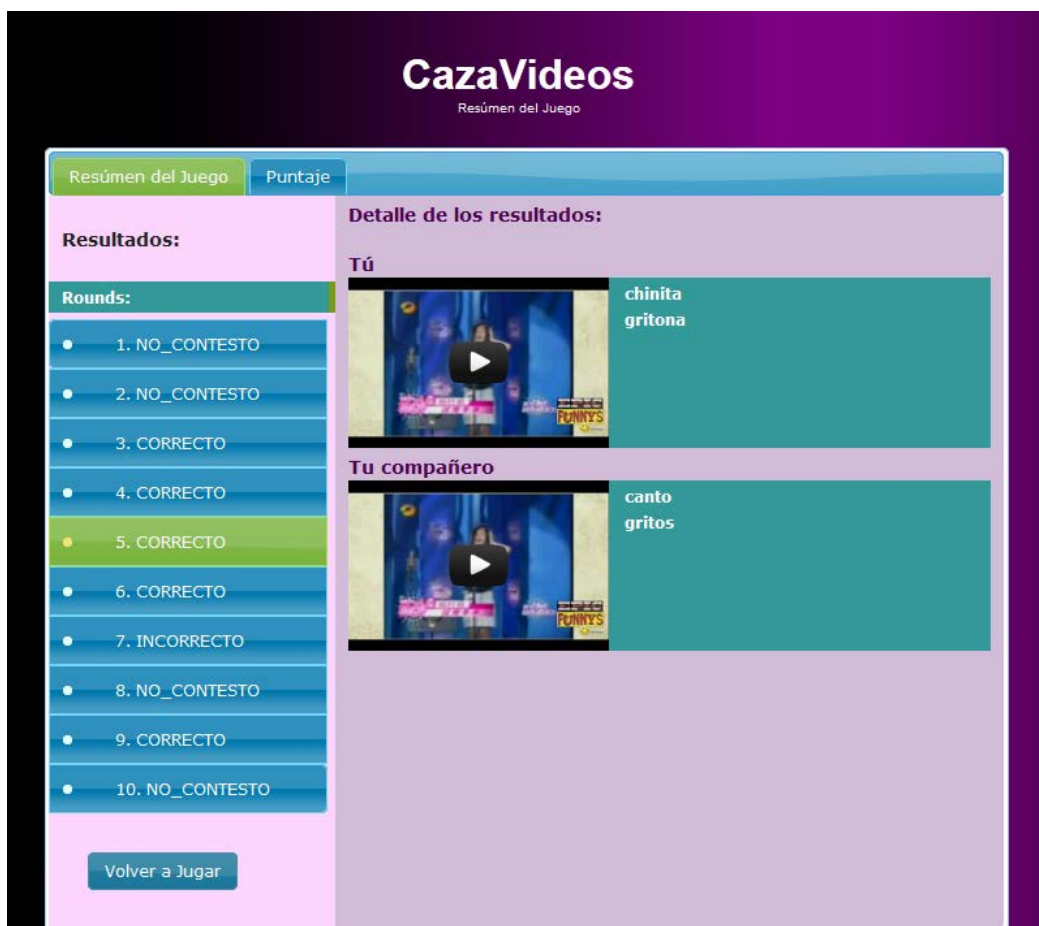


Figura 4. 8 - Resumen del Juego.

Las pruebas de funcionamiento de los módulos que implicaban sincronización, no resultaron favorables en su totalidad, debido a problemas con la librería Socket.io (ver Figura 4.9).

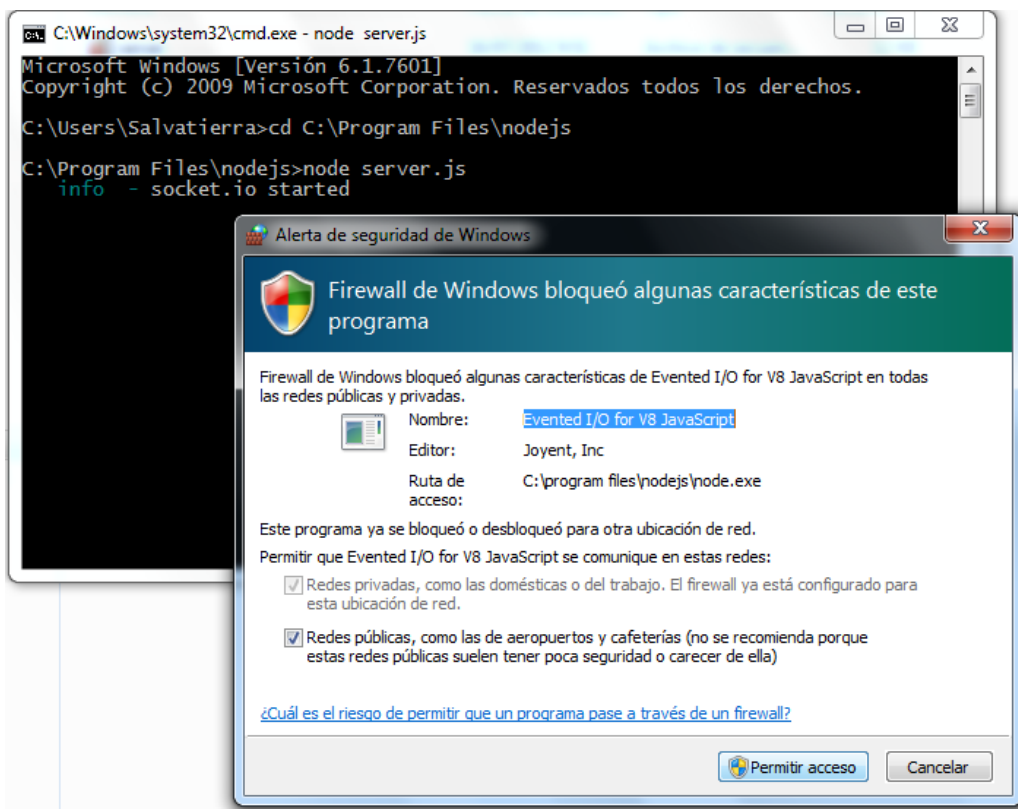


Figura 4. 9 - Módulo de Node.js - Socket.io

Luego de estas pruebas, se empezó a tomar en cuenta dos requisitos, no contemplados en el diseño, que son: asegurarse de que los dispositivos usados para probar mantengan usables los puertos destinados para la aplicación y el Sincronizador, y que se cumpla con el ancho de banda mínimo requerido para un funcionamiento aceptable, en este caso, 512Kbps.

Otra de las razones, que más tarde impidió concretar las sincronizaciones, fue el Error de tipo 100 del API de YouTube, el cual indicaba, que el video había sido removido por alguna razón o había sido marcado como privado, de tal manera que al hacer un requerimiento del mismo, no se encontraba; estos videos, en su mayoría de categorías no educativa, fueron remplazados.

4.5.2. Calidad de la información recolectada

Para la medición de la calidad de la información, de un total de 30 videos de la categoría “Educación”, y como se explicó en la sección 3.2.3, las etiquetas que se almacenaron corresponden a fragmentos de video de 30 segundos. Se tomaron en cuenta 36 fragmentos de dichos videos y se generó la nube de tags, un ejemplo de esto se muestra en las imágenes a continuación, donde la figura 4.10 muestra una clase de cálculo diferencial en el fragmento de tiempo entre los minutos 00:11:50 y 00:12:20. El tema principal tratado en este caso es “punto de inflexión”. Luego, en la figura 4.11 se puede observar la nube de tags generada como producto de las sesiones de juego invertidas específicamente en este ejemplo, y por último, en la 4.12 se muestran

estas mismas etiquetas luego del proceso de filtrado descrito en la sección 3.2.3.

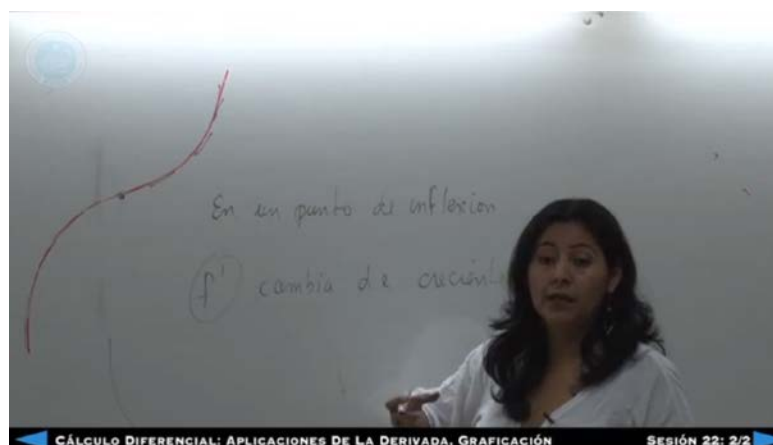


Figura 4. 10 - Fragmento de video entre 00:11:50 y 00:12:20

clase clases curvas derivada
diferencial grafica
inflexion pizarra
profesora punto

Figura 4. 11 - Etiquetas antes del filtrado

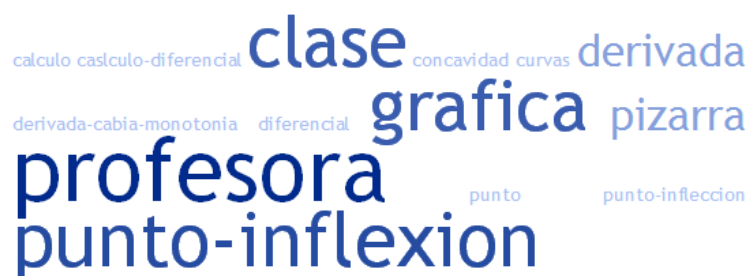


Figura 4. 12 - Etiquetas después del filtrado

De la figura 4.12 también se puede visualizar que el par de etiquetas más importantes es “profesora” y “punto-inflexión”. El número de jugadores que tuvieron que participar hasta llegar a obtener etiquetas válidas es de 5 en promedio. Este promedio se obtuvo consultando los datos de las sesiones almacenados en la base de datos de la aplicación (ver Anexo 1), donde se obtuvo el número de jugadores que etiquetaron cada fragmento hasta producir por lo menos dos etiquetas iguales relevantes. Este proceso se lo realizó offline para garantizar la agilidad de la aplicación durante su ejecución.

4.5.3. Análisis de resultados

Al observar las nubes de etiquetas de la figura 4.12, la etiqueta de mayor frecuencia es muy general, por tanto no agrega valor al

etiquetado del fragmento. Por otro lado, omitiendo esta, se puede observar que efectivamente se logró etiquetas válidas, aportando realmente una descripción al tema que se está desarrollando, como por ejemplo: “punto-inflexión”.

El número de usuarios necesarios resultó ser impar, debido a que no necesariamente los usuarios etiquetan en par el mismo vídeo. Teniendo en cuenta que el promedio de usuarios es de 5 y que los usuarios podrían tardar hasta 30 segundos en realizar las anotaciones, podemos decir que un usuario se tardaría alrededor de 2 minutos y 50 segundos en etiquetar un fragmento de vídeo; mientras un humano, para etiquetar ese mismo fragmento, debería ver todo el contenido.

Las estadísticas del Juego se muestran en la Tabla VIII.

Total etiquetas recolectadas:	5529
Total etiquetas únicas:	1514
Sesiones de juego:	210
Total de usuarios que participaron en la aplicación:	105

Tabla VIII – Estadísticas del Juego.

Las etiquetas consideradas válidas, que fueron obtenidas luego del procesamiento se muestran en la Tabla IX.

separable	punto	rectilíneo	sustitución
valor-absoluto	geometría	primer-orden	regla-cadena
factor-integrante	analítica	movimiento	integración-parte
Newton	cartesiano	parabólico	parte
cantidad-movimiento	producto-vectorial	caída	partes
Fricción	producto-punto	segundo-orden	tangente
factorial	producto-escalar	libre	área
derivada	técnicas-conteo	transpuesta	definida
cóncava	desplazamiento	antisimétricas	suma-telescópica
punto-inflexión	electroestática	matrices-cuadradas	transformada
reactivos	fuerza-magnética	matriz-identidad	laplace
simple-sustitución	aceleración	identidad	familia-curvas
estequiometria	aceleración	cuadradas	parábola
velocidad	negativa	determinante	ecuación-primer-orden
distancia	movimiento	rectilíneo	complementaria
separable	punto	determinante	transformada-z

Tabla IX – Etiquetas válidas.

En las siguientes tablas se muestra la cantidad de etiquetas obtenidas luego del procesamiento. En la Tabla X, aquellas que hemos denominado bloqueadas por no aportar con información específica del contexto del fragmento de video, y en la Tabla XI, la cantidad de etiquetas válidas, pertenecientes al intervalo 12, ver nube de etiquetas de la Figura 4.12.

Cantidad de etiquetas	Usuarios de Facebook
4	Usuario1
1	Usuario2
2	Usuario3
2	Usuario4

Tabla X – Cantidad de etiquetas bloqueadas del intervalo 12

Cantidad de etiquetas	Usuarios de Facebook
4	Usuario1
3	Usuario3
1	Usuario5

Tabla XI – Cantidad de etiquetas válidas por usuario del intervalo 12

CONCLUSIONES

1. Los GWAPs presentan un mecanismo eficaz para la obtención de etiquetas relevantes con cero costo en lo que se refiere a las personas que participan en el proceso y un costo mínimo de infraestructura para mantener la aplicación Web online.
2. Es posible que una aplicación consiga, con relativamente pocos usuarios, mejorar en 10500%⁴ el tiempo en que se realiza la generación de etiquetas para material audiovisual educativo comparado con la contratación de un experto.
3. Actualmente WebSocket es una alternativa para establecer una comunicación bidireccional entre cliente y servidor, es decir, para permitir

⁴ Donde 10500% se obtiene de los 12 minutos que participaron las 105 personas involucradas. Así: $(1260 \text{ minutos} * 100\%) / 12$

la construcción de aplicaciones en tiempo real. A este, se suman las ventajas de Node.js, por permitir que esa comunicación cliente-servidor se desarrolle en el mismo lenguaje y además sencillo, que es JavaScript.

4. Debido a que en el mecanismo aplicado, "Input-Agreement", se permite el ingreso de todo tipo de etiquetas, para no ser tan restrictivos y más bien velar por la motivación del jugador, es necesario realizar el procesamiento offline, para el filtrado de las etiquetas ingresadas.

5. Es importante que para evitar terminar con etiquetas muy generales, se filtren las etiquetas con respecto a la aparición de una palabra, de tal manera que la frecuencia de la misma en un determinado documento con respecto a la cantidad de documentos en los que aparece, refleje su importancia o ponderación real y permita así obtener las palabras más relevantes de un texto.

RECOMENDACIONES

1. Para futuras investigaciones en esta misma línea, se recomienda buscar métodos para medir el tiempo y las personas necesarias para generar etiquetas de todos los subtemas de un vídeo.
2. Profundizar en el análisis contextual de manera que, formas más específicas de vídeo puedan ser descritas para distintas industrias en idioma español.
3. En cuanto a la réplica de la aplicación se recomienda obtener videos de canales públicos o implementar un canal de YouTube propio del juego, así se podría evitar errores de tipo “100”⁵ de YouTube.

⁵ Error 100 - Según el API Reference de Youtube: Este error se produce cuando el vídeo solicitado no fue encontrado. Cuando un vídeo se ha eliminado (por cualquier razón) o se ha marcado como privado.

4. Experimentar con una mecánica de juego entre parejas no colaborativo, es decir, competitiva, ya que varios de los usuarios manifestaron que preferían recibir puntaje por “ganarle” a su pareja. Otra opción sería un híbrido colaborativo-competitivo, como lo implementa el juego “KissKissBan”.

5. Para la implementación, es aconsejable minimizar el paso de mensajes, una forma de hacer esto podría ser evitando el uso de otro framework como cliente, sino que se maneje la lógica directamente en Node.js; En ese caso se puede usar un framework para Node.js y la base de datos para no albergar demasiadas variables en memoria. Existen bases de datos no relacionales para Node.js como Redis, CouchDB, MongoDB que aportarían un acceso rápido a datos en el juego.

6. En cuanto a la generación de etiqueta final, se deja un camino trazado que, se recomienda, empezar mejorando el filtrado de la información, introduciendo técnicas de completado de raíces de palabras, como por ejemplo, usando Stemming Completion.

7. También utilizar una propia colección de Stop Words, que incluyan palabras generales.

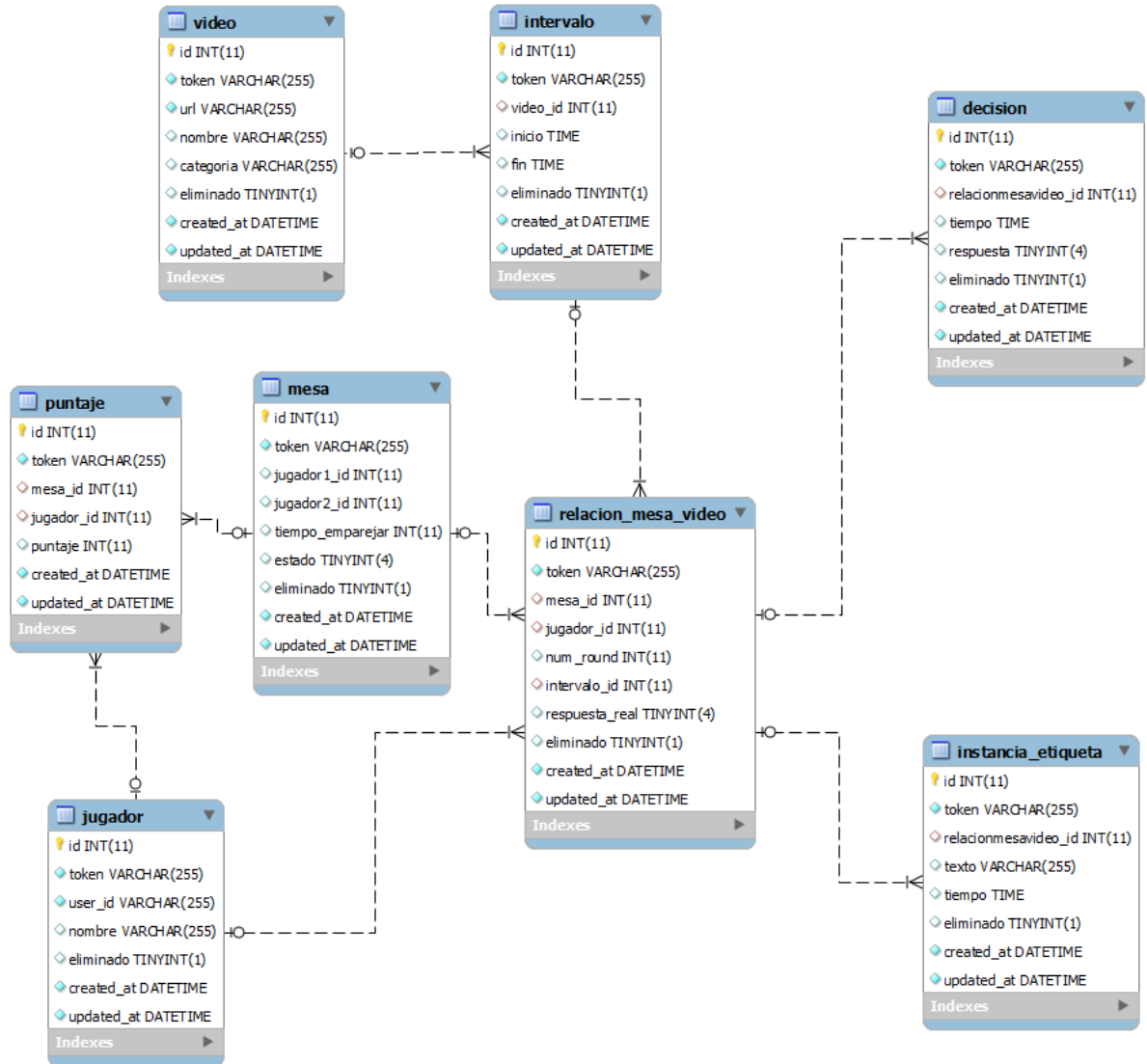
8. Para la frecuencia de las etiquetas, apuntar a maximizar la especificidad de las etiquetas de mayor frecuencia, para esto se podría hacer uso del cálculo de “tf-idf”, que es una medida estadística que refleja cuán importante es una palabra para un documento en una colección o corpus, y es usada como factor de ponderación en Recuperación de Información (Information Retrieval) y en Minería de Texto (Text Mining).

9. Para una puesta en producción se recomienda, especificar claramente las instrucciones y requisitos del juego.

ANEXOS

ANEXO 1

Modelo de la base de datos



ANEXO 2

Manual de Instalación

Esta sección contiene instrucciones para instalar y configurar la aplicación 'CazaVideos' en Windows.

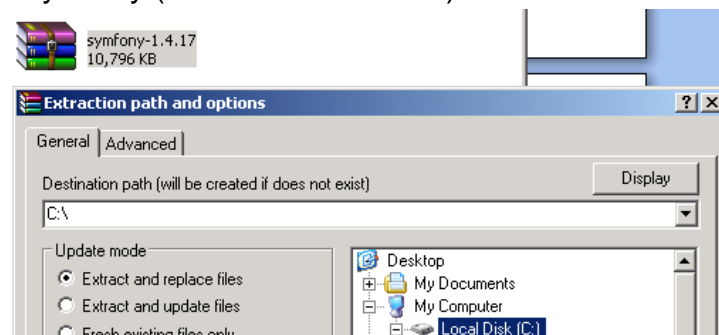
Requerimientos:

Instalar WampServer (PHP,MySQL), Symfony, node.js con el módulo socket.io.

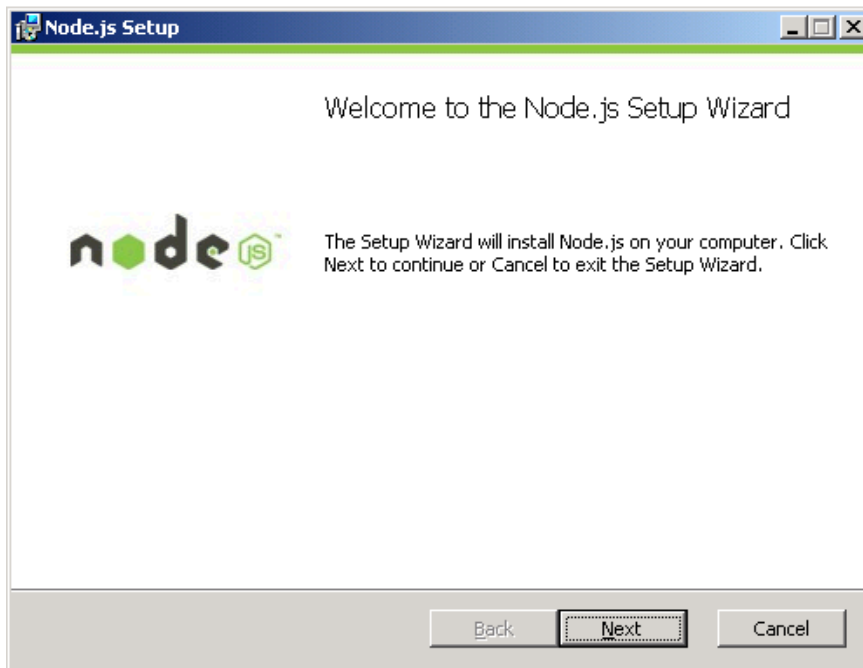
*WampServer (PHP,MySQL)



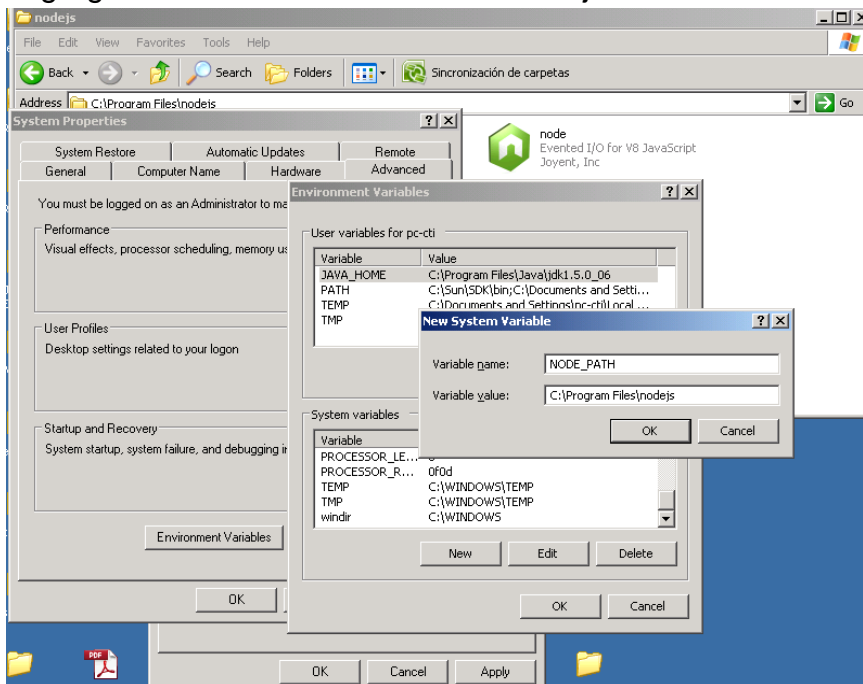
*Symfony (Extraer en disco C:\)



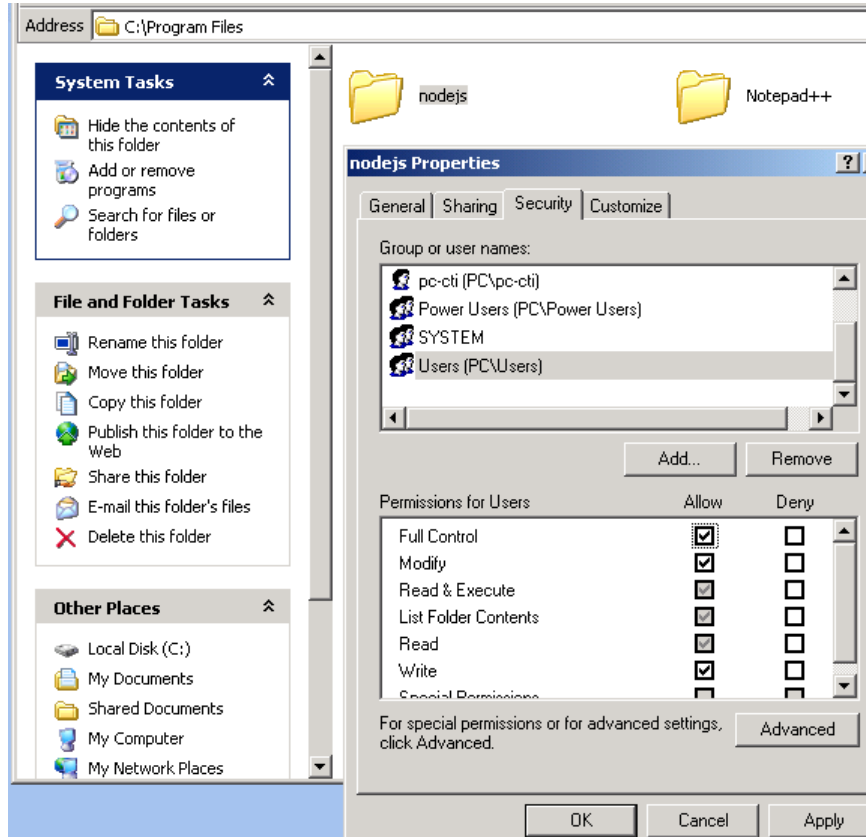
*Node.js



*Agregar la variable de entorno de node.js NODE_PATH.

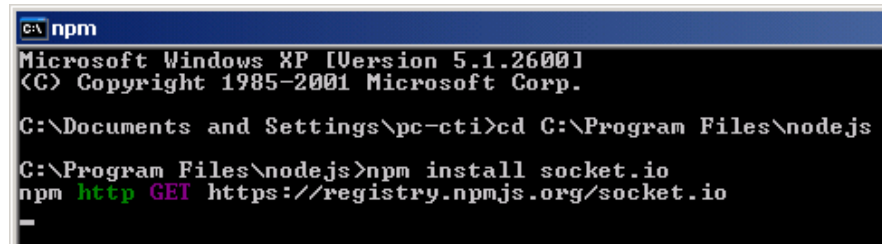


*Darle todos los permisos al directorio principal de node.js.



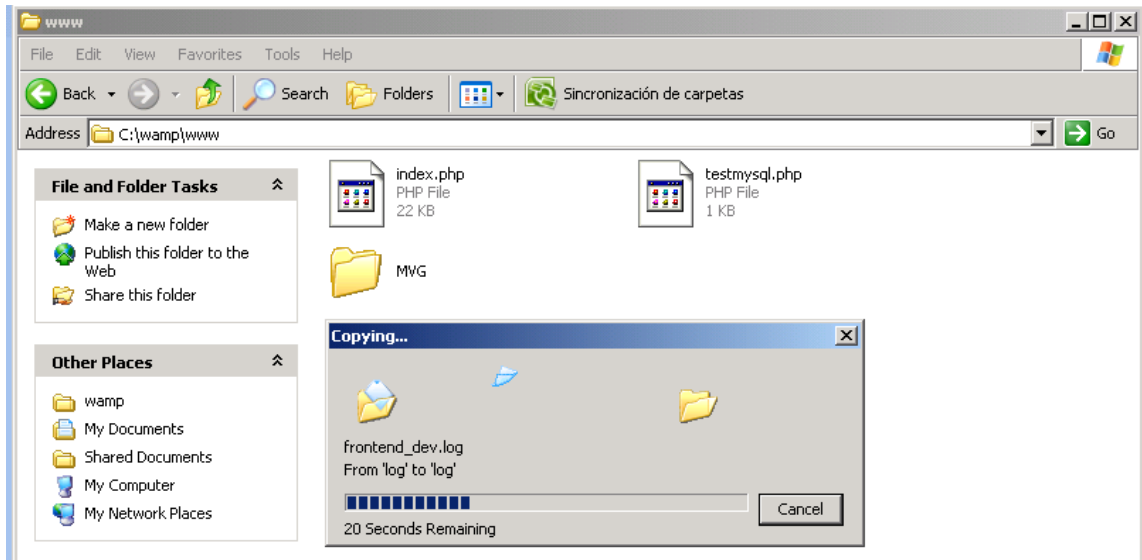
*Instalar socket.io con el gestor de paquetes incluido en el mismo node.js:

```
npm install socket.io
```

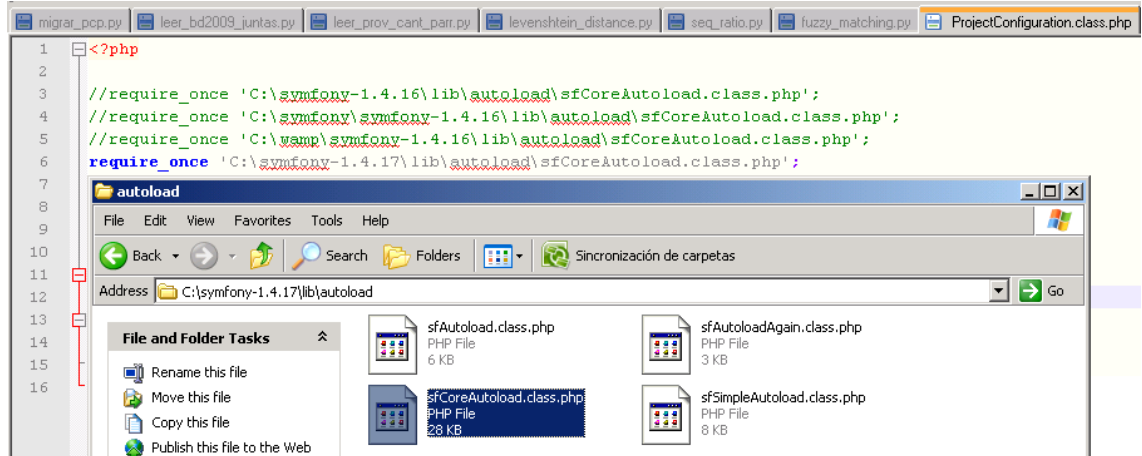


Configuración:

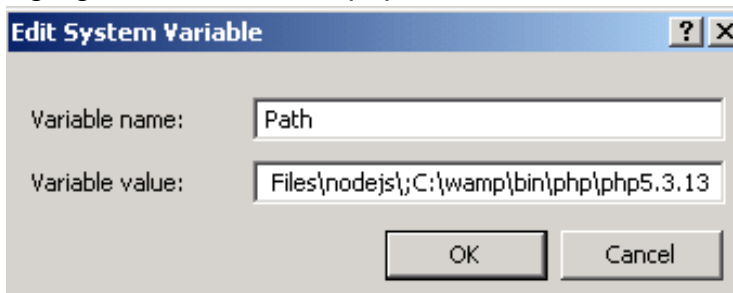
Colocar la aplicación en el directorio www de WAMP.



Modificar el archivo ProjectConfiguration.class.php

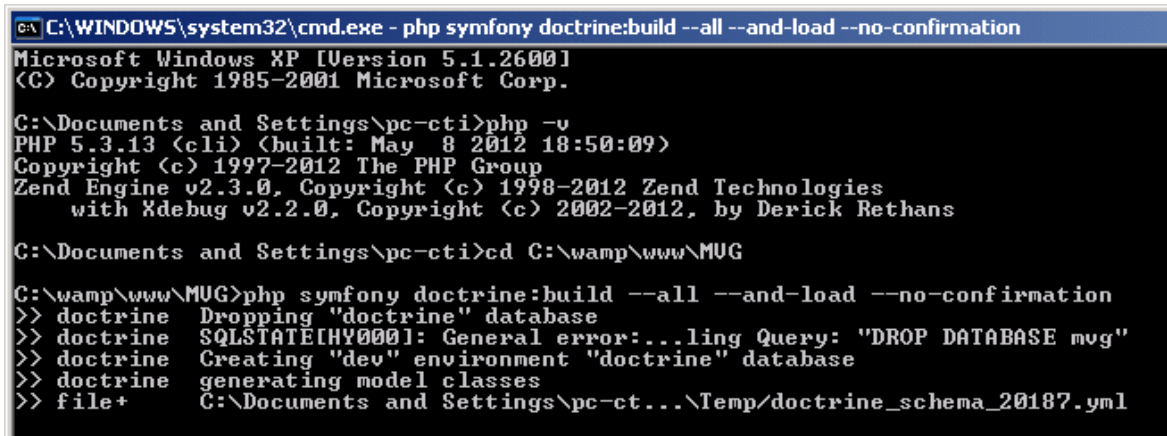


Agregar el directorio de php a la variable de entorno Path:



Para configurar 'CazaVideos', debe ejecutar el siguiente comando de Symfony, en el directorio principal de la aplicación, para que todo sea generado del esquema de la base (modelos, validadores, formularios y se recreen todas las tablas y se carguen los datos iniciales).

```
php symfony doctrine:build --all --and-load --no-confirmation
```



```
CAV C:\WINDOWS\system32\cmd.exe - php symfony doctrine:build --all --and-load --no-confirmation
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\pc-cti>php -v
PHP 5.3.13 (cli) (built: May 8 2012 18:50:09)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.3.0, Copyright (c) 1998-2012 Zend Technologies
    with Xdebug v2.2.0, Copyright (c) 2002-2012, by Derick Rethans

C:\Documents and Settings\pc-cti>cd C:\wamp\www\MUG

C:\wamp\www\MUG>php symfony doctrine:build --all --and-load --no-confirmation
>> doctrine  Dropping "doctrine" database
>> doctrine  SQLSTATE[HY000]: General error:...ling Query: "DROP DATABASE mug"
>> doctrine  Creating "dev" environment "doctrine" database
>> doctrine  generating model classes
>> file+    C:\Documents and Settings\pc-ct...\Temp/doctrine_schema_20187.yml
```

Adicionalmente, remover todo lo que se haya almacenado en el directorio cache:

```
php symfony clear-cache    o    php symfony cc
```

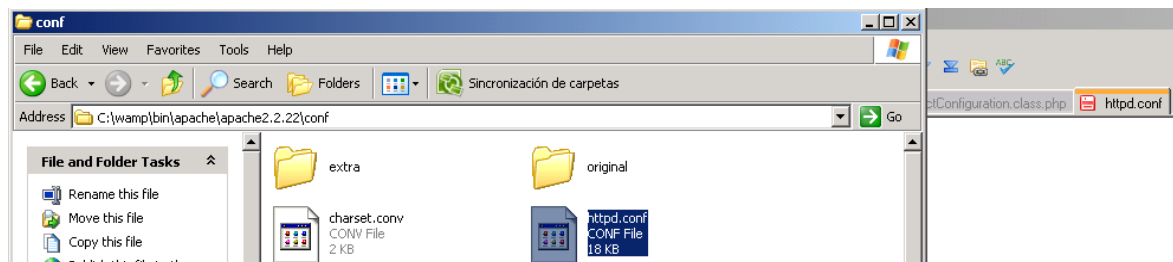
```

C:\WINDOWS\system32\cmd.exe
C:\wamp\www\MUG>php symfony cc
>> cache      Clearing cache type "all" for "frontend" app and "dev" env
>> file+      C:\wamp\www\MUG\data/frontend_dev-cli.lck
>> chmod 777  C:\wamp\www\MUG\data/frontend_dev-cli.lck
>> file-      C:/wamp/www/MUG/cache/frontend/...odules_Mesa_config_view.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...es_Mesa_config_security.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...ules_Mesa_config_module.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...les_Mesa_config_filters.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...dules_login_config_view.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...s_login_config_security.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...les_login_config_module.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...es_login_config_filters.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/.../config/config_settings.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/dev/config/config_routing.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...config/config_factories.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/...config/config_databases.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/.../config/config_handlers.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/.../config/config_auto_load.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/dev/config/config_app.yml.php
>> file-      C:\wamp\www\MUG\data/frontend_dev-cli.lck
>> cache      Clearing cache type "all" for "frontend" app and "prod" env
>> file+      C:\wamp\www\MUG\data/frontend_prod-cli.lck
>> chmod 777  C:\wamp\www\MUG\data/frontend_prod-cli.lck
>> file-      C:/wamp/www/MUG/cache/frontend/.../config/config_settings.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/.../config/config_handlers.yml.php
>> file-      C:/wamp/www/MUG/cache/frontend/prod/config/config_app.yml.php
>> file-      C:\wamp\www\MUG\data/frontend_prod-cli.lck

C:\wamp\www\MUG>

```

Modificar el archivo httpd.conf del servidor de aplicaciones apache, y agregar esta configuración y reiniciar todos los servicios:



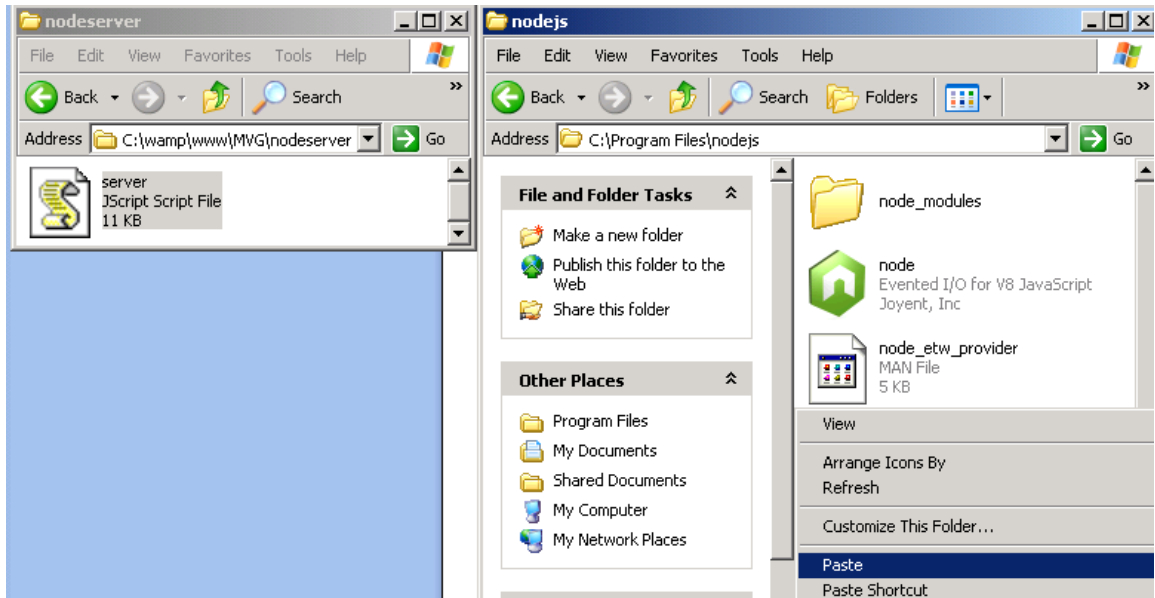
```

<VirtualHost 127.0.0.1>
ServerName musicvideogame.localhost
DocumentRoot "C:\wamp\www\MVG\web"
DirectoryIndex index.php
<Directory "C:\wamp\www\MVG\web">
AllowOverride All
Allow from All
</Directory>

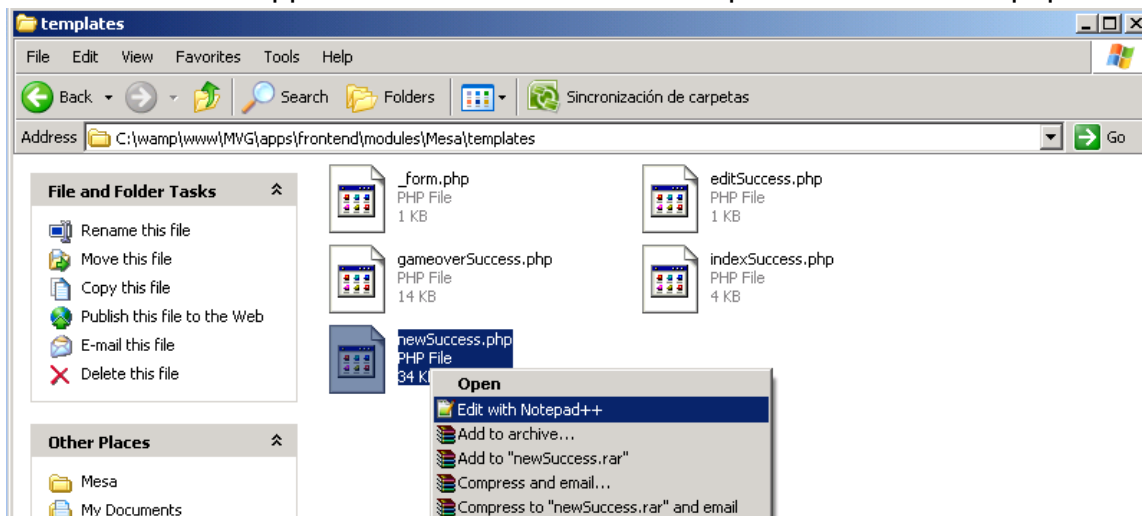
Alias /sf "C:\symfony-1.4.17\data\web\sف"
<Directory "C:\symfony-
1.4.17\data\web\sف">
AllowOverride All
Allow from All
</Directory>
</VirtualHost>

```

Colocar el archivo server.js en el directorio donde se encuentre instalado node.js.



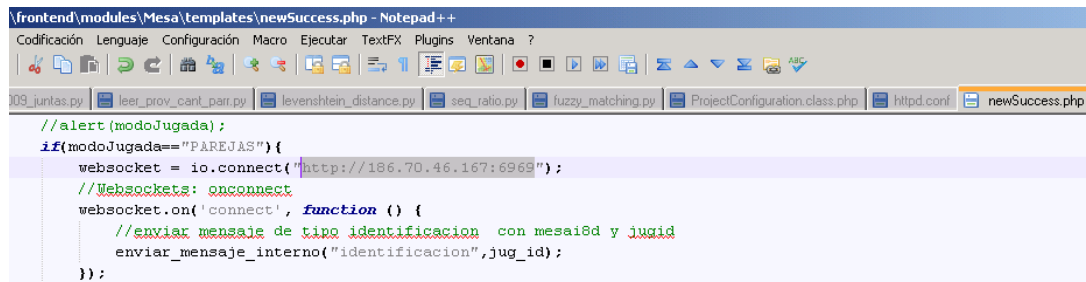
Modificar la url (**ip:puerto**) indicada en la plantilla inicial de la aplicación, ubicada en MVG/apps/frontend/modules/Mesa/templates/newSuccess.php:



```
<script src="http://190.154.171.111:6969/socket.io/socket.io.js">
</script>
```

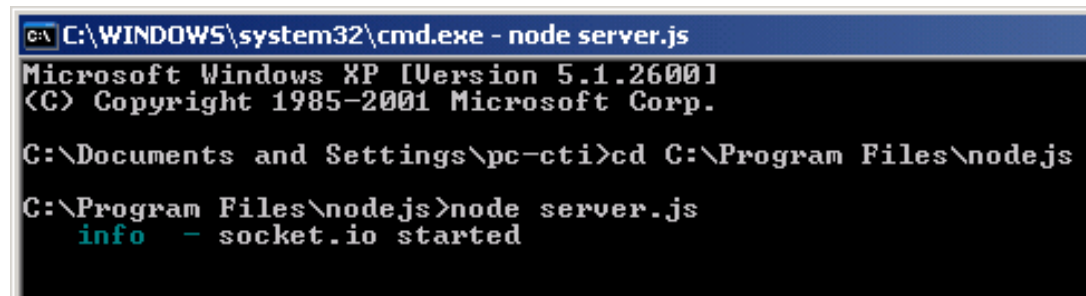
Y en:

```
websocket = io.connect("http://190.154.171.111:6969");
```

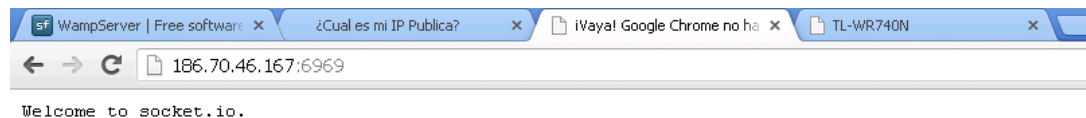


```
//alert(modoSjugada);  
if(modoSjugada=="PAREJAS"){  
    websocket = io.connect("http://186.70.46.167:6969");  
    //Websockets: onconnect  
    websocket.on('connect', function () {  
        //enviar mensaje de tipo identificacion con mesaId y jugid  
        enviar_mensaje_interno("identificacion",jug_id);  
    });  
}
```

Levantar node.js, y acceder a esta url indicada. Ahí deberá devolver un mensaje "socket.io", para verificar si socket.io está escuchando por ese puerto. Levantar el servidor de aplicaciones apache, o en este caso iniciar el WampServer (ambiente de desarrollo que contiene a Apache2, PHP y MySQL). Finalmente, probar la aplicación en 2 navegadores diferentes.

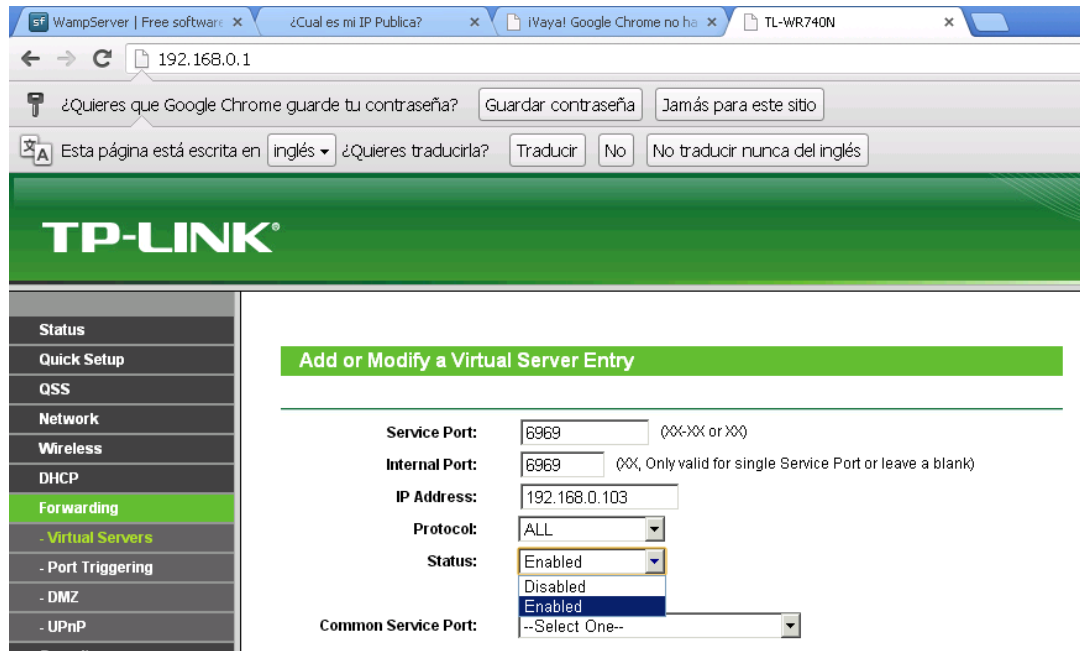


```
C:\WINDOWS\system32\cmd.exe - node server.js  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\Documents and Settings\pc-ctid>cd C:\Program Files\nodejs  
C:\Program Files\nodejs>node server.js  
info - socket.io started
```

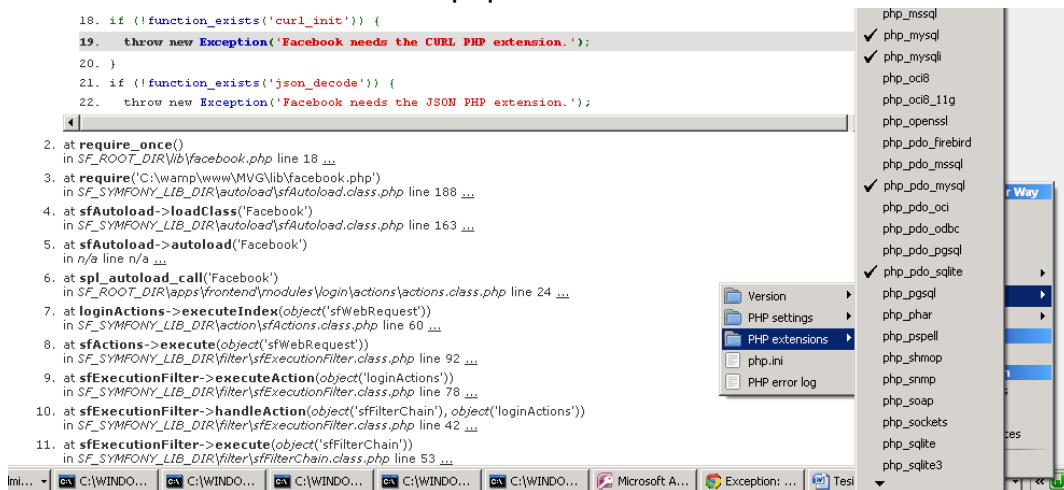


Observación:

*Recuerde configurar el router para habilitar tanto el puerto 80 como el 6969, usados en esta aplicación.



*Recuerde activar la extensión php_curl.



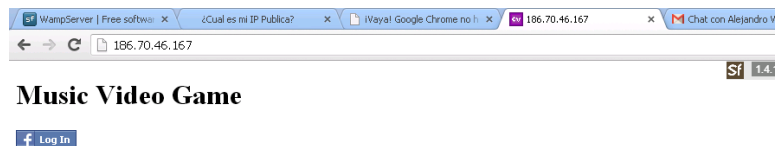
*Modificar la configuración de la aplicación en Facebook Developers.

The screenshot shows the Facebook Developers app configuration page for an application named 'MVGAME'. The app ID is 424542557563209 and the app secret is 1e028adabc5079f46e187c550930bc67. The app is currently live. The 'Información básica' section includes fields for Display Name (MVGAME), Namespace, Contact Email (taws@espol.edu.ec), App Domains, and Hosting URL. The 'Sandbox Mode' is set to 'Opción desactivada'. The 'Selecciona cómo tu aplicación se entrega con Facebook' section shows 'Website with Facebook Login' selected, with the site URL set to 'http://186.70.46.167/index.php/Mesa?'. A sidebar on the left contains navigation links for developer roles, app details, review status, open graph, localization, alerts, insights, and related links.

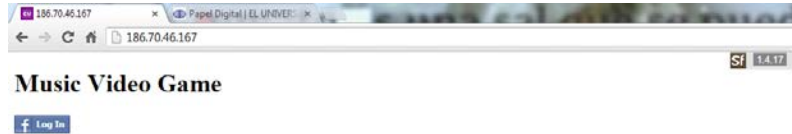
Pruebas de la aplicación

Esta sección contiene las pruebas previas al funcionamiento.

1. Comprobar la creación de parejas de juego o asignación de compañeros de juego.
 - a. Ingresar a la aplicación desde 2 navegadores diferentes.
 - b. Ingresar con 2 usuarios de Facebook, diferentes.



Jugador A

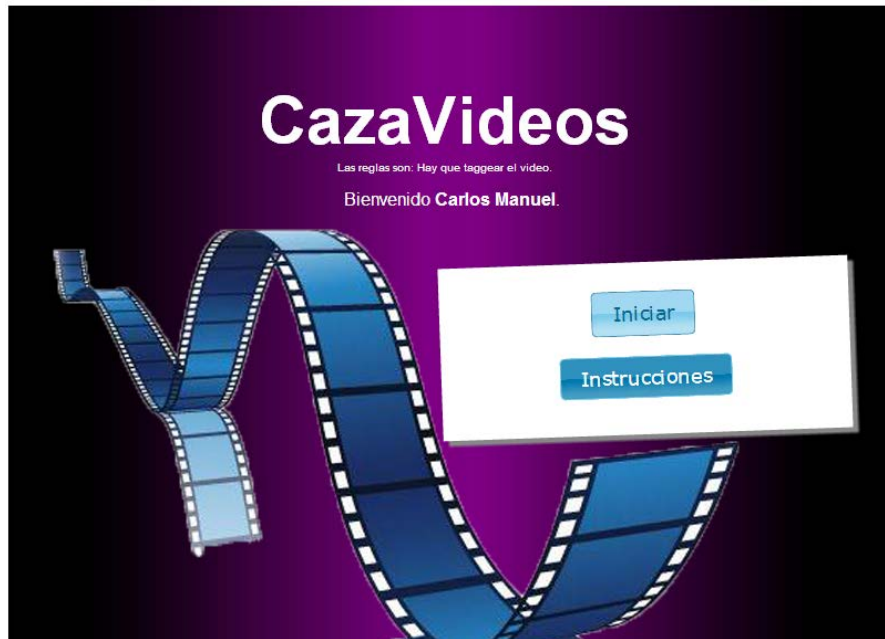


Jugador B

- c. Iniciar partida coincidiendo con un compañero de juego. (Click en botón Iniciar).

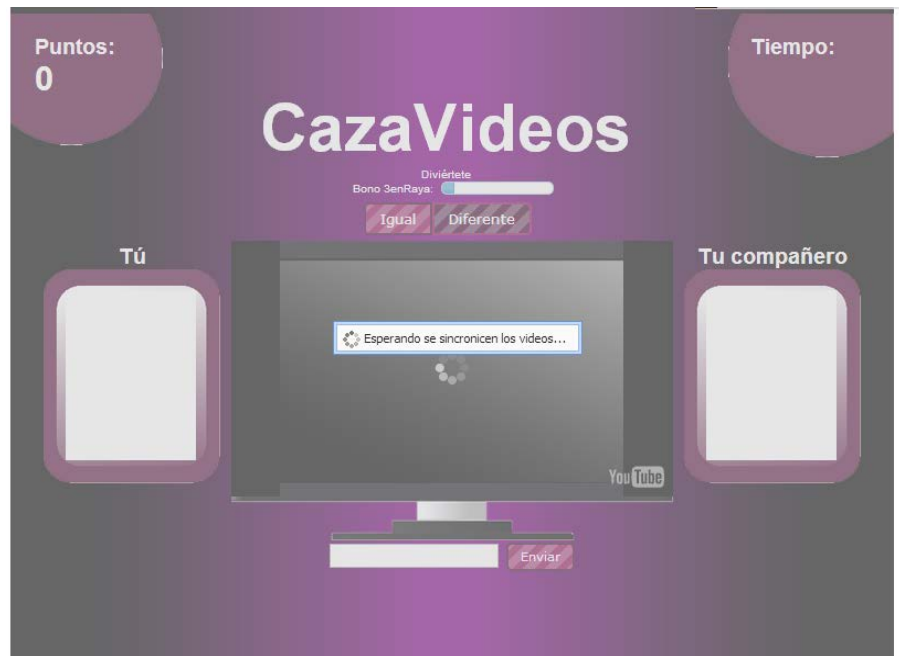


Jugador A



Jugador B

2. Determinar el tiempo de respuesta en aquellas partes de la aplicación, que implican sincronización, como:
 - **Asignación de fragmentos de videos a cada uno de los jugadores.** (Verificar en YouTube, el acceso permitido a esos videos asignados).
 - a. Una vez que se han creado las parejas de jugadores. Espere a que terminen de sincronizar la reproducción de videos de los jugadores.
 - b. Comprobar que haya finalizado la sincronización en ambos navegadores, y que se estén reproduciendo ambos videos.



- **La sección de intercambio de mensajes.**

- a. Una vez que se estén reproduciendo los videos en ambos navegadores, comprobar el funcionamiento de la sección de chat desde ambos navegadores (envíe un mensaje cualquiera a su compañero). Compruebe la recepción en el otro navegador.



Jugador A



Jugador B

- **Intercambio de respuestas con los botones Igual o Diferente.**
 - a. Antes de que acabe la ronda en cuestión, dar click en el botón que indica que el video es "Igual" al de su compañero, o al botón que indica que es "Diferente" (envíe su respuesta). Compruebe la recepción en el otro navegador.

BIBLIOGRAFÍA

- [1] Von Ahn, L. Human computation. Tesis Doctoral. UMI Order Number: AAI3205378, CMU, 2005
- [2] Surowiecki, James. The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday, 2004.
- [3] Von Ahn, L. y Dabbish, L. General Techniques for Designing Games with a Purpose. Communications of the ACM, Agosto 2008. Páginas 58-67.
- [4] Von Ahn, L. y Dabbish, L. Labeling images with a computer game. In Proc. SIGCHI Conf. on Human Factors in Computing Systems (Vienna, April 24-29). ACM, New York, 319-326, 2004.
- [5] Von Ahn, L., Liu, R., and Blum, M. Peekaboom: A Game for locating objects in images. In Proceedings of the SIGCHI Conference on Human

Factors in Computing Systems (Montreal, Apr. 22–27). ACM Press, New York, 2006, 55–64.

- [6] Von Ahn, L., Kedia, M., and Blum, M. Verbosity: A game for collecting common-sense knowledge. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montreal, Apr. 22–27). ACM Press, 2007, 75–78.
- [7] Law, E., von Ahn L., Dannenberg, R., and Crawford M. TagATune: a game for music and sound annotation. In Proc. 8th Intl. Conf. on Music Information Retrieval (Vienna, September 23-27). 361-364, 2007.
- [8] Von Ahn, L. "Games with a Purpose," IEEE Computer Magazine, vol. 39, no. 6, pp. 92-94, June 2006.
- [9] Siorpaes K., Hepp M., "Games with a Purpose for the Semantic Web," IEEE Intelligent Systems, vol. 23, no. 3, pp. 50-60, May-June 2008.

- [10] Law, E. y von Ahn, L. Input-Agreement: A New Mechanism for Collecting Data Using Human Computation Games. ACM Conference on Human Factors in Computing Systems, CHI 2009. Páginas 1197-12
- [11] Turing, A. I.—computing machinery and intelligence. *Mind*, 1950, LIX (236), 433–460.
- [12] Grier, D. *When Computers Were Human*. Princeton University Press, 2005, 411 p.
- [13] Yang Yang, Zhu Bin B., Guo Rui, Yang Linjun, Li Shipeng, and Yu Nenghai. 2008. A comprehensive human computation framework: with application to image labeling. In *Proceedings of the 16th ACM international conference on Multimedia (MM '08)*. ACM, New York, NY, USA, 479-488.
- [14] Yuen, M., Chen, L., & King, I. *A Survey of Human Computation Systems*. IEEE Symposium on Social Computing Applications (SCA'09, in conjunction with IEEE SocialCom'09), Vancouver, Canada, 2009.

- [15] Mandel, M. I., & Ellis, D. P. W. A Web-Based Game for Collecting Music Metadata. (S. Dixon, D. Bainbridge, & R. Typke, Eds.) Journal of New Music Research, 2008, 37(2), 151-165. Österreichische Computer Gesellschaft.
- [16] Quinn A. J. y Bederson B. B. Human computation: A survey and taxonomy of a growing field. In Proceedings of CHI 2011, 2011, Vancouver, BC, Canada.
- [17] Howe, J. The Rise of Crowdsourcing. Wired Magazine, 2006, 14(6), 1-5
- [18] Law, E. y von Ahn, L. Human Computation. Morgan & Claypool Synthesis Lectures on Artificial Intelligence and Machine Learning, 2011.
- [19] Herdagdelen A. y Baroni M. The concept game: Better commonsense knowledge extraction by combining text mining and a game with a purpose. In AAI Fall Symposium on Commonsense Knowledge, 2010.
- [20] Ho, C.-ju, Chang, T.-hsuan, Lee, J.-chuan, Hsu, J. Y.-jen, & Chen, K.-ta. (2009). KissKissBan: A Competitive Human Computation Game for Image

Annotation Categories and Subject Descriptors. Memory, 1-4. ACM Press.

Recuperado de <http://portal.acm.org/citation.cfm?id=1600153>

[21] Gneezy U. y Rustichini A. Pay enough or don't pay at all. Quarterly journal of economics, 2000, 115 (3), 791–810,

[22] Ubl, M. y Kitamura, E. (2010, Octubre 20) Introducing Websockets: Bringing Sockets to the Web. Recuperado de: <http://www.html5rocks.com/en/tutorials/websockets/basics/>