

# INFORME

## de Preprocesado y Minería de Datos



Apoya



Cofinanciado por el  
programa Erasmus+  
de la Unión Europea

# **Informe de preprocesado y minería de datos**

Resultado del proyecto ACACIA (561754-EPP-1-2015-1-CO-EPPKA2-CBHE-JP) cofinanciado por el programa Erasmus+ ACACIA: Centros de Cooperación para el Fomento, Fortalecimiento y Transferencia de Buenas Prácticas que Apoyan, Cultivan, Adaptan, Comunican, Innovan y Acogen a la comunidad universitaria.

**Autores:**

Fabián José Roldán Piñeros (UNINOVA).

**Coordinadores del proceso de validación:**

Olga Lucia León Corredor, (UDFJC)

John Paez, (UDFJC)

**Editor:**

Pedro Enrique Espitia Zambrano

**Diseño gráfico y diseño editorial accesible:**

Pedro Enrique Espitia Zambrano

Esta obra se distribuye bajo Licencia Creative Commons Atribución No Comercial Sin Derivar 4.0 Internacional



## Agradecimientos

Esta obra ha sido posible gracias al compromiso de todos los miembros del consorcio del proyecto ACACIA (Centros de Cooperación para el Fomento, Fortalecimiento y Transferencia de Buenas Prácticas que Apoyan, Cultivan, Adaptan, Comunican, Innovan y Acogen a la comunidad universitaria) y a la cofinanciación recibida por parte de la Comisión Europea a través del Programa Erasmus+.

El consorcio del proyecto ACACIA está conformado por tres universidades europeas y once de América Latina:

Universidad Distrital Francisco José de Caldas (UDFJC), Colombia. Universidad Nacional de Educación a Distancia (UNED), España. Universidade Nova de Lisboa (UNINOVA), Portugal. Pontificia Universidad Católica de Valparaíso (PUCV), Chile. Universidad Federal do Oeste do Pará (UFOPA), Brasil. Universidad Nacional Mayor de San Marcos (UNMSM), Perú. Universidad de Antofagasta (UA), Chile. Universidad de

las Regiones Autónomas de la Costa Caribe Nicaragüense (URACCAN), Nicaragua. Universidad Pedagógica Nacional (UPN - Colombia), Colombia. Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP), Brasil. Universitatea “1 Decembrie 1918” Alba Iulia (UAB), Rumania. Corporación Universitaria Iberoamericana (CUI), Colombia. Universidad Nacional Autónoma de Nicaragua, León (UNAN – León), Nicaragua. Universidad Continental (UC), Perú. INCI: Instituto Nacional para Ciegos de Colombia, Colombia. INSOR: Instituto Nacional para Sordos de Colombia, Colombia. Fundación Sidar Acceso Universal (de ámbito iberoamericano).



## Contenido

1. About this document.....	2
Historial de cambios.....	2
Datos del documento.....	3
2. Lista de Abreviaturas.....	4
3. Resumen.....	4
4. Arquitectura de Software (SAD).....	4
4.1 Vista Lógica.....	4
4.2 Vista de Implementación.....	6
4.3 Vista de Procesos.....	7
4.3.1 Paso de recopilación de datos.....	7
4.3.2 Paso de extracción de datos.....	8
4.3.3 Paso de pre-proceso.....	8
4.3.4 Paso de generación de modelo.....	10
4.3.5 Paso de validación & Uso.....	10
4.4 Vista de Física.....	10
5. Conclusiones.....	14
6. Referencias.....	14



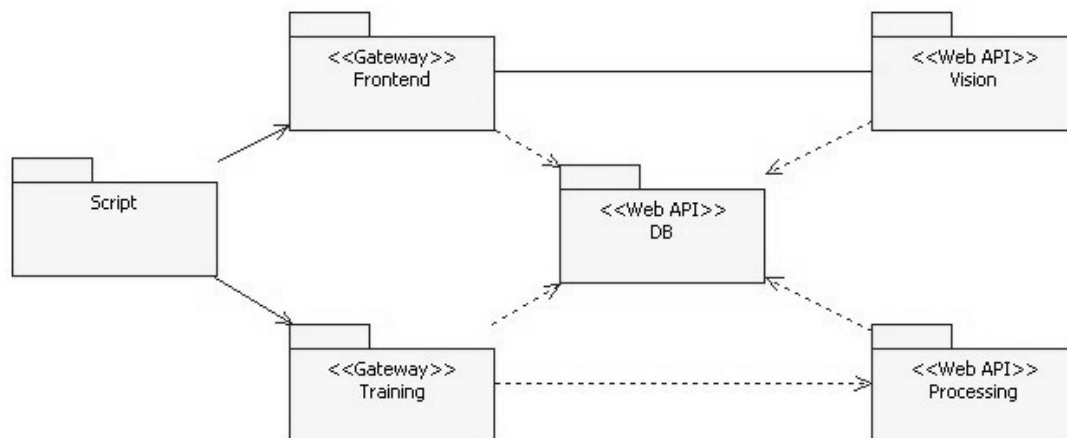
SAD *Software Architecture Document*: Documento de especificación de arquitecturas de software.

Este documento describe los contenidos del entregable E.5.1.3 titulado “Informe Técnico Parte II”. El entregable está compuesto por una descripción técnica del prototipo de sistema recomendador. El presente documento se refiere técnicamente hablando a un extracto del documento SAD (*Software Architecture Document*).

En el presente documento se efectúa una descripción técnica de la arquitectura de software del prototipo (en adelante sistema) con base al modelo de descripción de soluciones de software modelo 4+1 diseñado por Philippe Kruchten (Kruch, 1995). A continuación se inicia la descripción de la arquitectura iniciando con la vista lógica del modelo 4 + 1 Vistas.

#### 4.1 Vista Lógica

La vista lógica describe al sistema desde el punto de vista estructural en donde se denotan los componentes y sus relaciones. A partir del método de diseño basado en patrones (Tekin, 2009). Se procedió a identificar el patrón de nivel de arquitectura base de micro-servicios (Richa, 2019) y se articuló el sistema de acuerdo a las exigencias computacionales de los requerimientos. En la Figura 1 se puede apreciar la vista lógica de nivel 0 (mayor nivel de abstracción del sistema):



**Figura 1. Vista lógica del sistema del patrón base de micro-Servicios.**

De acuerdo a diseño del patrón de micro-servicios el aislamiento de componentes es una característica importante a tener en cuenta, sin embargo como se puede apreciar en el diseño se optó por la consideración de usar un Web API para prestar servicios de acceso a datos de persistencia de manera común a los demás componentes ya que el repositorio de datos del sistema funge el rol de un *Data Warehouse*. En la siguiente tabla se especifica cada componente lógico del sistema:

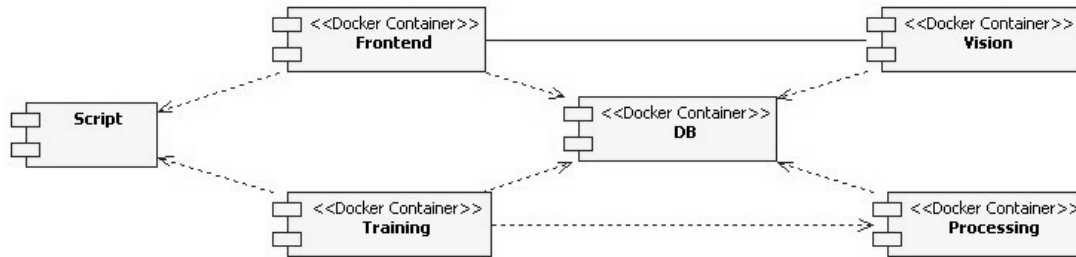
ID CP	Nombre	Descripción
CP01	<i>Script</i>	<p>Se trata de un componente de cliente ligero que se incorpora a la plataforma EDX bajo la figura de <i>plugin</i>. Es el componente en cargado de las siguientes tareas:</p> <ul style="list-style-type: none"> <li>▪ Capturar la aceptación o denegación de los términos de uso por parte del usuario Estudiante.</li> <li>▪ Capturar a través de la cámara web del dispositivo fotografías del estudiante mientras que presenta el curso con una frecuencia de 30 segundos.</li> <li>▪ Desplegar la información de resultados de desempeño de los estudiantes para un curso en específico.</li> <li>▪ Mostrara las alertas generadas por el sistema.</li> <li>▪ Mostrar las recomendaciones generadas por el sistema.</li> </ul>
CP02	<i>Frontend</i>	Gateway el cual interactúa con el componente script ofreciéndole los servicios asociados a la aceptación de los términos de uso y la captura de fotografías.

		En el momento que recibe una nueva captura fotográfica, serializa la imagen, luego, se comunica con el componente DB para el almacenamiento de la imagen y luego se comunica con el componente Vision para iniciar el proceso de imagen.
CP03	<i>Vision</i>	Web API encargada de procesar las imágenes que le indica el componente Frontend. El componente de visión genera un vector de características el cual es almacenado en el componente DB.
CP04	<i>Training</i>	Gateway el cual interactúa con el componente script ofreciéndole los servicios desplegar información de resultados de desempeño de los estudiantes, mostrar alertas y recomendaciones bajo un esquema <i>PUSH</i> generadas por el sistema hacia los profesores. Adicionalmente permite al usuario administrador controlar la máquina de aprendizaje automático (Componente Processing).
CP05	<i>DB: Data Warehouse</i>	Web API encargada de mantener el <i>Data Warehouse</i> del sistema.
CP06	<i>Processing</i>	Web API encargada de realizar el proceso de entrenamiento y clasificación de la máquina de aprendizaje automático.

## 4.2 Vista de Implementación

La vista de implementación describe como los componentes lógicos son materializados con el uso de herramientas tecnológicas. Describe la configuración de estas herramientas en términos de lenguaje de programación empleado y versión. Lo anterior comúnmente se conoce como configuración de componentes de software.

El patrón de micro-servicios usualmente se implementa sobre una plataforma tecnológica que a nivel de despliegue faculta a los diferentes componentes de la arquitectura un aislamiento para potenciar la seguridad, tolerancias a fallas, desempeño y mantenimiento del sistema. La tecnología seleccionada para tal fin fue *Dockers* (Babac, 2017). Cada componente con excepción del componente Script se implementa dentro de un contenedor *docker* como se puede apreciar en la siguiente Figura 2:



**Figura 2. Vista de implementación del sistema.**

Cada componente de lógica corresponde uno a uno con un componente de implementación, por lo tanto los componentes mantienen el mismo ID de especificación para mantener la trazabilidad del sistema. En la siguiente tabla se especifica las tecnologías empleadas de los componentes:

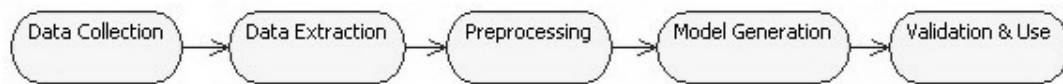
ID CP	Configuración			
	Imagen	Lenguaje	Versión	Herramientas
CP01	Ninguna	Javascript	1.1	Web Socket
CP02	Ubuntu	Javascript	1.0	Node.js, Express
CP03	Debian	C++	1.1	Afectiva SDK
CP04	Ubuntu	Python	1.0	Django
CP05	Ubuntu	SQL	1.0	Postgres
CP06	Ubuntu	Java SE	1.0	WEKA, Hibernnet, Pluzly

### 4.3 Vista de Procesos

La vista de procesos describe las rutinas de ejecución que emplea el sistema para la prestación de sus funcionalidades y los procesos lógicos que interviene en esas rutinas. El sistema hace uso de técnicas de aprendizaje de máquina para identificar las conductas negativas de los estudiantes cuando están desarrollando un curso. Para la identificación de la conducta la presente investigación se baso en el modelo de Ejes de Expresión Emocional (Chara, 2013) en donde a partir del reconocimiento de emociones sumado a las expresiones faciales del estudiante se clasifica su estado comportamental en las clases de: aburrido, ansioso y concentrado.

Por lo tanto se requirió entrenar una máquina de aprendizaje para que dispusiera con la capacidad de reconocer el estado comportamental entre las clases del modelo de Ejes de Expresión Emocional a partir de la información capturada de fotografías del estudiante a medida del desarrollo del curso. En la Figura 3 se puede apreciar el Procesos de Entrenamiento que se empleo de acuerdo a los lineamientos por Steven Samelson (Samel, 2012):



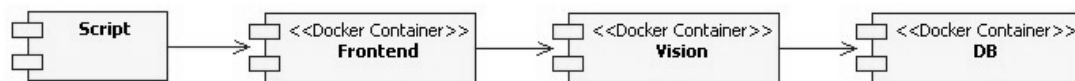


**Figura 3. Proceso de entrenamiento de máquina de aprendizaje.**

Como se puede apreciar en la Figura 3 se ilustra el proceso de entrenamiento de la máquina de aprendizaje que consta de cinco pasos que van desde la obtención de los datos hasta la validación y uso de la máquina de aprendizaje resultante. A continuación, se describen las principales actividades realizadas en cada paso del proceso de entrenamiento de la máquina de aprendizaje.

### 4.3.1 PASO DE RECOPIACIÓN DE DATOS

El objetivo del paso de recopilación es de fungir como un puente entre las fuentes datos y el componente *data warehouse* personalizado del sistema.



**Figura 4. Componentes relacionados con la extracción de datos.**

En la Figura 4 se puede apreciar los componentes relacionados con la extracción de datos. El componente *Script* con una frecuencia de 30 segundos captura fotografías del estudiante las cuales son enviadas al servidor y recibidas por el componente *Frontend*. El componente *Frontend* almacena la imagen serializada en *data warehouse*. Luego el componente *Frontend* le envía una señal al componente de *Vision* quien efectúa una identificación del estado emocional del individuo a partir del reconocimiento facial de emociones y expresiones. Como resultado de la identificación de características el vector resultante tiene la siguiente forma:

TimeStamp,faceId,interocularDistance,glasses,age,ethnicity,gender,dominantEmoji,pitch,yaw,roll,joy,fear,disgust,sadness,anger,surprise,contempt,valence,engagement,smile,innerBrowRaise,browRaise,browFurrow,noseWrinkle,upperLipRaise,lipCornerDepressor,chinRaise,lipPucker,lipPress,lipSuck,mouthOpen,smirk,eyeClosure,attention,eyeWiden,cheekRaise,lidTighten,dimpler,lipStretch,jawDrop,relaxed,smiley,laughing,kissing,disappointed,rage,smirk,wink,stuckOutTongueWinkingEye,stuckOutTongue,flushed,scream,

### 4.3.2 PASO DE EXTRACCIÓN DE DATOS

El objetivo del proceso de extracción de datos es de establecer un modelo de representación datos para efectos de entrenamiento. De acuerdo a los lineamientos de Hadley Wickham para la consecución de un *Tidy Data Set* se adoptaron las siguientes reglas de transformación (Wickh, 2017):

- Cada variable conforma una componente del vector de características resultante.
- El conjunto de datos resultante se debe normalizar en el rango de reales entre [0, 1].
- Los literales categóricos de una variable discreta debe ser parte de una componente del vector de características resultante. Y los valores deben corresponder a valores booleanos 0 o 1 según sea el caso.
- Para aquellos conjuntos de datos cuya combinación (producto cruz) es significativa el vector resultante debe contener estas componentes normalizadas.

Por lo tanto, el componente *Processing* obteniendo el vector características del data *warehouse* le efectúa las reglas de transformación obteniendo el vector de características normalizado.

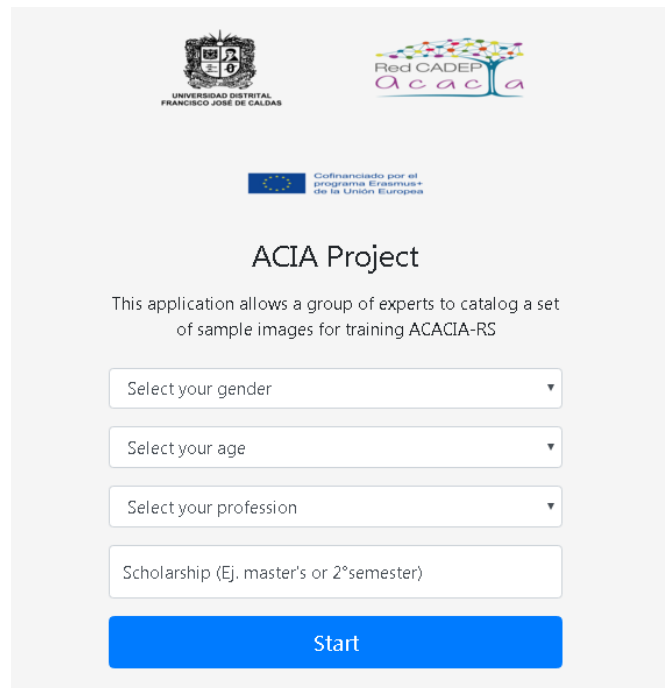
### 4.3.3 PASO DE PRE-PROCESO

El objetivo del paso de pre-procesamiento de datos es acondicionar el conjunto de datos para el proceso de entrenamiento conforme a la técnica de clasificación seleccionada. Para efectos del presente trabajo se selecciono el algoritmo de clasificación K-MEANS siguiendo una estrategia de aprendizaje de máquina no supervisada. Con ayuda del componente externo WEKA (Mark, 2003). Una vez acondicionado el conjunto de datos a las instancias de objeto de WEKA se procedió a dividir el conjunto de datos de instancia en los siguientes sub-conjuntos:

- **Conjunto de Datos de Entrenamiento:** 80% del conjunto total de datos para identificación de los *clusters* aplicando el algoritmo K-MEANS con los siguientes parámetros:

Parámetro	Valor
Cantidad de <i>Cluster</i>	3
Data	Instancias de Entrenamiento
Iteraciones	1000

- **Conjunto de Datos de Validación Interna:** 10% del conjunto total de datos. Este conjunto de datos también es conocido como conjunto de datos de prueba para calcular el error cuadrático medio respecto a la distancia entre la predicción (etiqueta) de la instancia y valor esperado.
- **Conjunto de Datos de Asignación:** 40% del conjunto total de datos. Este conjunto corresponde a la mitad del conjunto de entrenamiento y se dispuso para proveerle significado a los *cluster* con respecto al concepto impartido por expertos. Para tal fin se dispuso de una aplicación web como se ilustra en la Figura 5:



**Figura 5. Captura de asignación de etiqueta a *clusters* por expertos.**

A cada experto se le dispuso de un conjunto de imágenes en donde ellos en su concepto reconocían la clase correspondiente y procedían a etiquetar. A partir de estos datos se generó un espacio de frecuencias los cuales se correlacionaron con los *clusters* denotándoles de esta manera su significado.

- **Conjunto de Datos de Validación Externa:** 10% del conjunto total de datos. Finalmente la validación externa se efectuó con el concepto de los expertos respecto a la el significado dado al correspondiente etiqueta al la instancia perteneciente a grupo de datos de validación externa.

#### 4.3.4 PASO DE GENERACIÓN DE MODELO

A partir de los conjuntos de entrenamiento y validación interna se efectúa la generación de *Clusters* con ayuda del componente WEKA. Una vez generados los *Clusters* el sistema procede a almacenar el modelo con el propósito de usarlo para clasificar durante la ejecución de la máquina respecto a nuevas instancias. Este proceso es llevado a cabo por el componente *Processing*.

#### 4.3.5 PASO DE VALIDACIÓN & USO

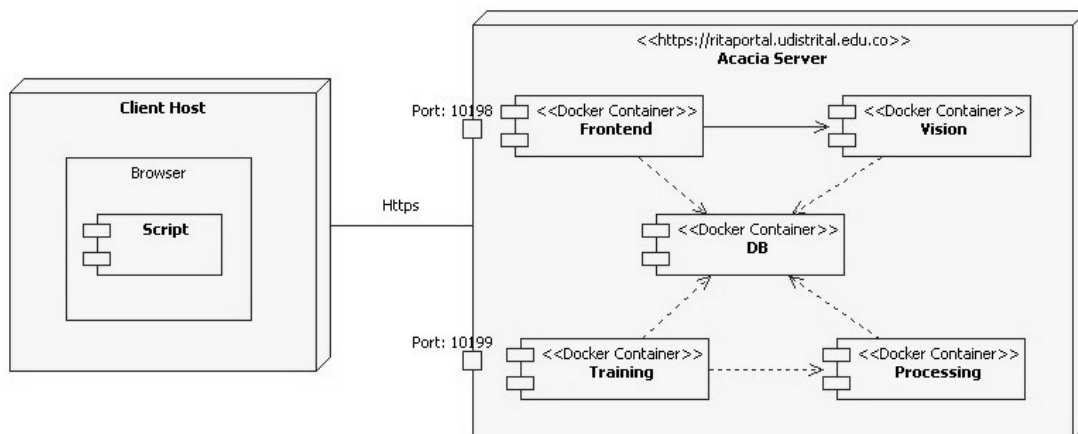
Para validación de *clusters* existen dos tipos de validación a saber: la validación interna y la validación externa. La validación interna esta en función de las mediadas de cohesión y separación de los clusters. Los cuales a través de la suma de cuadrados y cálculo del error cuadrático medio, respectivamente dan una idea de la calidad de los clusters generados, esta información es brindada

por el componente WEKA tras la ejecución del proceso de generación del conjunto de validación interna.

La validación externa cuando se conoce de antemano la etiqueta o *cluster* correspondiente a una instancia se procede a compara la salida de la máquina de aprendizaje con respecto a la salida esperada. Dando como resultado un índice de aceptabilidad de resultados de máquina para tal fin se empleo el conjunto de validación externa cuyas instancias se conocía de antemano su cluster correspondiente gracias al dictamen del conjunto de expertos. Finalmente el modelo queda almacenado en el sistema de archivos local del componente *Processing* para su carga y ejecución (clasificación de nuevas instancias).

#### 4.4 Vista de Física

Como el nombre lo indica la vista física describe la ubicación de los componentes de implementación dentro del sistema. En esta vista se describen características de componentes como topología, servidores, medios de comunicación, protocolos y esquemas de despliegue. En la Figura 6 se puede apreciar la topología del sistema junto a una especificación técnica de los componentes físicos involucrados:



**Figura 6. Vista física del sistema.**

A continuación se especifica los componentes físicos ilustrados en la Figura 6:

ID AT	Artefacto	Descripción
AT01	<i>Client Host</i>	Corresponde al dispositivo del cliente que puede ser un portátil o un computador de escritorio. El nodo es independiente del tipo de usuario y consta del componente Script el cual es.

		<p>desplegado a través de un navegador web. Los navegadores aptos son:</p> <ul style="list-style-type: none"> <li>▪ Chrome versión superior o igual a 75.0.3770.</li> <li>▪ Firefox versión superior o igual a 61.0a1.</li> </ul> <p>El javascript debe estar habilitado en los navegadores y el dispositivo debe tener mino 2 megas de RAM 1.7 de procesador y conexión a internet.</p>
AT02	<i>Acacia Server</i>	<p>Corresponde al lado del servidor, en donde consta de los demás componentes de implementación. Consiste en un servidor en donde se expone dos puertos a través de https de los puntos de enlace de los gateway respectivamente: Frontend (10198) y Training (10199); las especificaciones técnicas de la maquina son:</p> <ul style="list-style-type: none"> <li>▪ Sistema Operativo: Ubuntu 16.04.</li> <li>▪ Memoria RAM: 8GB.</li> <li>▪ 4 CPU de 2GHz.</li> <li>▪ 33 GB de espacio libre.</li> </ul>

En la siguiente sección se describen las pautas a tener en cuenta para el despliegue del sistema con respecto a ser implementada con una plataforma de cursos en línea como EDX.

### **Pautas**

El sistema se sirve de un *plugin* (Componente Script) para capturar las imágenes de los estudiantes a medida que presentan el curso. Una vez que el sistema cuenta con la información de los estados emocionales genera una serie de alertas y recomendaciones las cuales se pueden apreciar por medio del servicio de obtención del comportamiento emocional. Las alertas y las recomendaciones pueden ser retroalimentadas a través del mecanismo de retroalimentación con el fin de mejorar las respuestas del sistema.

#### **1. Activar la captura de emociones:**

Para activar la captura de emociones se requiere por cada sección donde se precise la captura de imágenes, adicionar una sección de tipo texto en donde en el editor HTML se debe insertar el script que se muestra a continuación:

### Código fuente HTML

```

1 <p></p>
2 <div style="display: none;"><video id="player" controls="controls" autoplay="autoplay"></video>
  <canvas id="snapshot" width="320" height="240"></canvas> <canvas id="image_for_crop"
  width="320" height="240"></canvas>
3 </div id="aja" style="width: 100px; height: 100px;"></div>
4 <script src="https://ritaportal.udistrital.edu.co:10198/js/acacia-plugin.js?v=1"></script>
5 </div>
6 <p></p>

```

El script usa los componentes de medios del navegador para el acceso al video cámara del dispositivo el cual bajo un periodo de 30 segundos captura las imágenes del estudiante mientras presenta el curso. Las imágenes, por tratarse de información personal requiere del consentimiento informado de la captura y tratamiento de los datos. Para este fin, se requiere crear una sesión de términos de uso cuyo script asociado se muestra a continuación:

### Código fuente HTML

```

1 <p>El presente curso se sirve de una tecnología novedosa que ayuda al tutor a brindar un
  mejor apoyo durante la ejecución del curso. La tecnología hace uso de la
  cámara de tu dispositivo para identificar tus estados emocionales y a
  partir de ellos recomendarle al tutor acciones de mejora durante el proceso de aprendizaje. Al
  dar clic en aceptar, le concedes al sistema el permiso de capturar imágenes mientras
  presentas el curso en línea. Estas imágenes son usadas para efecto de
  la realización de recomendaciones por parte del sistema.</p>
2 <div>
3 <div id="button-box"><input type="button" class="button" value="Aceptar"
  onclick="sendAuthorization(true);" /> <input type="button" class="button" value="Declinar"
  onclick="sendAuthorization(false);" /></div>
4 <script src="https://ritaportal.udistrital.edu.co:10198/js/acacia-validator.js?v=2"></script>
5 </div>
6 <p></p>

```

## 2. Activar servicio de obtención del comportamiento emocional:

Para activar el servicio de obtención del comportamiento emocional se requiere crear una sección oculta a los estudiantes en donde se incorpore el siguiente script en el editor HTML:

EDITOR CONFIGURACIÓN

```

Editando: Tablero
1 <style>.link-box{margin: 22px auto; width: 80%; text-align: center;}</style>
2 <div class="link-box"></div>
3 <script>
4   var courseID = document.getElementsByClassName('course-number')[0].textContent;
5   var linkbox = document.getElementsByClassName('link-box')[0]
6   var btn = document.createElement("button");
7   var url = 'http://ritaportal.udistrital.edu.co:10199/get-control-board?course=' + courseID;
8   btn.innerHTML = "Ver Resultados";
9   btn.onclick = function() {
10    window.open(url);
11  };
12  linkbox.appendChild(btn);
13 </script>

```

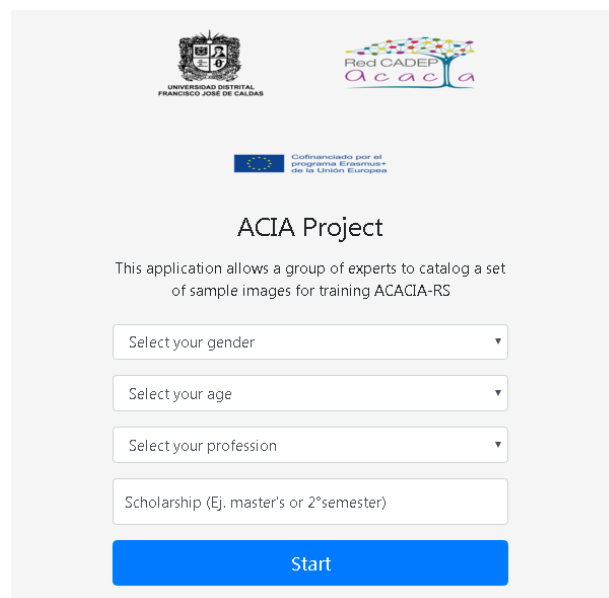
Al dar clic en ver resultados el navegador muestra una nueva pestaña en donde se puede apreciar la información consolidada de desempeño de los estudiantes del curso, como también el comportamiento:

### Cuadro de Control

Alias	Inicio	Frecuencia	Estado
Josealbert	Year: 119	95	Active
PaulF	Year: 119	120	Active
CaterinePardo	Year: 119	565	Active
pmacarena	Year: 119	1	Active
JohnPa	Year: 119	4	Active
Williams	Year: 119	87	Active
daniellaramirez	Year: 119	66	Active

### 3. Retroalimentación del sistema:

El sistema contempla dos mecanismos de captura de información referente a claridad de los resultados que pueda emitir. En primer lugar dispone de un mecanismo de entrenamiento el cual expone un subconjunto de imágenes para que puedan ser clasificadas por usuarios expertos. Esta clasificación permite al sistema aprender la correspondencia de las imágenes en los conjuntos del modelo. Para acceder al mecanismo de entrenamiento se dispone del recurso Web: <http://ritaportal.udistrital.edu.co:10199/> . En este recurso se solicita datos generales (no personales) del experto y en él se muestra una serie de imágenes susceptibles a ser catalogadas.



The screenshot shows the ACIA Project registration page. At the top, there are logos for Universidad Distrital Francisco José de Caldas and Red CADEF Acacia. Below these is a banner for the Erasmus+ program. The main heading is "ACIA Project". The text below reads: "This application allows a group of experts to catalog a set of sample images for training ACACIA-RS". There are four input fields: "Select your gender", "Select your age", "Select your profession", and "Scholarship (Ej. master's or 2º semester)". A blue "Start" button is at the bottom.

El segundo mecanismo consiste en un sistema de calificación por parte del docente sobre la información presentada por el sistema. En la parte inferior del detalle se le presenta al docente un conjunto de estrellas las cuales de izquierda a derecha indican el nivel de conformidad por parte del docente con los resultados presentados por el sistema frente al estudiante actual.



El sistema se sirve de una red neuronal que supervisa y ajusta los resultados entregados por el sistema. Cuando un conjunto similar de resultados supera un umbral de tolerancia (varios docente sobre un mismo patrón indican estar desacuerdo) la red neuronal reactiva la generación del modelo de aprendizaje incorporando este nuevo conjunto de resultados como un aprendizaje (el patrón no pertenece a la clase que se estaba indicando inicialmente).

### **Red Neuronal**

Para ajustar las respuestas del sistema a medida que los usuarios Profesores vayan usando la plataforma. Se implemento una red neuronal con base de un combinador lineal adaptativo de una capa oculta con la siguiente configuración:

Como se puede observar en la imagen se dispone de tres neuronas de entrada y tres neuronas de salida por cada uno de los conjuntos de clasificación. Y existen tres neuronas de adaptación las cuales reciben la alimentación proveniente de los profesores en la entrada deseada (datos de entra codificados). De manera lineal la salida final del sistema se va ajustando de acuerdo a la modulación que aplica las neuronas de adaptación.

## 5. Conclusiones

Este documento describe la arquitectura del sistema recomendador. Es denotar el carácter experimental del sistema y por consiguiente el sentido de prototipo que reviste. Se empleo la estructura de micro-servicios como patrón base. Revistiendo al sistema un carácter distribuido para soportar la carga computacional que significa el procesar de imágenes. Se uso una estrategia no supervisada de aprendizaje de maquina empleando el algoritmo K-MEANS para la segmentación y clasificación de instancias.



## 6. Referencias

Babak Bashari Rad (2017). An Introduction to Docker and Analysis of its Performance. Asia Pacific University of Technology and Innovation Technology Park Malaysia, Kuala Lumpur, Malaysia.

Charalampous and Kokkinos (2013). Charalampous, K. and Kokkinos, C. M. (2013). The model of interpersonal teacher behaviour: a qualitative cross-cultural validation within the greek elementary education context. *British Educational Research Journal*, 39(1):182{205.

Kruchten, Philippe (1995, November). Architectural Blueprints — The “4+1” View Model of Software Architecture., *IEEE Software*.

Mark, Hall (2003). The WEKA Data Mining Software: An Update. Department of Computer Science. Suite 340, 5950 Hazeltine National Dr. University of Waikato. Orlando, FL 32822, USA Hamilton, New Zealand

Richardson, Chris (2019). Arquitectura de micro-servicios. Recurso Web tomado el 13 de mayo de 2019: <https://microservices.io/patterns/microservices.html>

Samelson, Steven (2012). Machine Learning: The Absolute Complete Beginner’s Guide to Learn and Understand Machine Learning From Beginners, Intermediate, Advanced, To Expert Concepts.

Tekin Erdogan, Befir (2009). Workshops and Symposia at MODELS 2009, Denver, CO, USA, October 4-9, 2009. ... Alfonso Pierantonio, Ella Roubtsova, Pablo Sánchez.

Wickham, Hadley (2017). Tidy Data, Rstudio - *Journal of Statistical Software*.