

**ANEXOS - ARQUITECTURA BASADA EN CONTEXTO PARA EL SOPORTE
DEL SERVICIO DE VOD DE IPTV MÓVIL, APOYADA EN SISTEMAS DE
RECOMENDACIONES Y STREAMING ADAPTATIVO**



M.Sc. GABRIEL ELÍAS CHANCHÍ G.

Tesis de Doctorado en Ingeniería Telemática

**Director:
PhD. José Luis Arciniegas H.**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de investigación en servicios avanzados de telecomunicaciones
Popayán, Noviembre de 2017**

GABRIEL ELÍAS CHANCHÍ GOLONDRINO

**ARQUITECTURA BASADA EN CONTEXTO PARA EL
SOPORTE DEL SERVICIO DE VOD DE IPTV MÓVIL,
APOYADA EN SISTEMAS DE RECOMENDACIONES Y
STREAMING ADAPTATIVO**

**Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de:**

**Doctor en:
Ingeniería Telemática**

**Director:
PhD. José Luis Arciniegas Herrera**

Popayán, 2017

Contenido

Lista de Figuras	iii
Anexo A.....	5
A. Generación del dataset de video	5
A.1 Proceso de generación dataset.....	5
Anexo B.....	9
B. Instalación y manejo de servidor Spark.....	9
B.1 Creación servicio básico.....	9
Anexo C	15
C. Instalación y manejo de servidor Flask.....	15
C.1 Instalación ejecución Flask	15
Anexo D	19
D. Manejo plataforma de Arduino Yún.....	19
D.1 Servicio web con Arduino Yún.....	19
Anexo E.....	23
E. Herramienta Weka 3.6	23
E.1 Clasificador bayesiano con Weka 3.6.....	23
Anexo F.....	29
F. Herramienta apache benchmark	29
F.1 Manejo apache benchmark	29
Anexo G	31
G. Generación de contenidos DASH-WebM.....	31
G1. Generación flujos de video y audio	31
G2. Indexación de contenidos	32
G3. Generación del archivo MPD.....	33

G4. Proceso de codificación automática	34
Anexo H	37
H. Servicio de VoD DASH	37
F.1 Manual de usuario	37
Anexo I	41
I. Prueba de usabilidad películashoy.com.....	41
I1. Instrumento de usabilidad	41
Anexo J	49
J. Registros software	49
J1. Registros software realizados	49

Lista de Figuras

<i>Figura A. 1. Librerías utilizadas y llave de Echonest.....</i>	<i>5</i>
<i>Figura A. 2. Búsqueda de canciones más populares EchoNest.....</i>	<i>6</i>
<i>Figura A. 3. Obtención de las características del contenido</i>	<i>6</i>
<i>Figura A. 4. Obtención de la emoción del contenido.....</i>	<i>6</i>
<i>Figura A. 5. Método para la obtención de url asociada.....</i>	<i>7</i>
<i>Figura A. 6. Método TinyDB y generación de archivo JSON</i>	<i>7</i>
<i>Figura A. 7.Descarga de contenidos multimedia</i>	<i>8</i>
<i>Figura B. 1. Creación proyecto maven.....</i>	<i>10</i>
<i>Figura B. 2. Agregar dependencia de Spark al proyecto</i>	<i>11</i>
<i>Figura B. 3. Ejemplo HelloWorld en Spark</i>	<i>13</i>
<i>Figura B. 4. Desplegar ejemplo HelloWorld.....</i>	<i>14</i>
<i>Figura C. 1. Instalación de Python en Linux.....</i>	<i>15</i>
<i>Figura C. 2. Instalación de librería Flask.....</i>	<i>16</i>
<i>Figura C. 3. Instalación del entorno de desarrollo Geany.....</i>	<i>16</i>
<i>Figura C. 4. Proyecto HelloWorld en Flask.....</i>	<i>16</i>
<i>Figura C. 5. Ejecución del proyecto HelloWorld en Flask.....</i>	<i>17</i>
<i>Figura D. 1. Conexión a la red arduino yún</i>	<i>19</i>
<i>Figura D. 2. Interfaz de configuración arduino yún.....</i>	<i>20</i>
<i>Figura D. 3. Figura D.3. Código prueba de arduino yún</i>	<i>21</i>
<i>Figura D. 4. Ejemplo de Hola mundo a través de la tarjeta arduino yún</i>	<i>21</i>
<i>Figura E. 1. Archivo arff weka.....</i>	<i>23</i>
<i>Figura E. 2. Explorador herramienta weka</i>	<i>24</i>
<i>Figura E. 3. Explorador herramienta weka</i>	<i>24</i>
<i>Figura E. 4. Filtros de weka</i>	<i>25</i>
<i>Figura E. 5. Algoritmos de clasificación weka</i>	<i>25</i>
<i>Figura E. 6. Opciones de salida. Fuente: propia.....</i>	<i>26</i>
<i>Figura E. 7. Análisis de la herramienta weka sobre los datos iniciales. Fuente: propia</i>	<i>26</i>
<i>Figura F. 1. Lanzando el servidor.....</i>	<i>29</i>
<i>Figura F. 2. Comando apache-benchmark</i>	<i>30</i>
<i>Figura F. 3. Despliegue de resultados.....</i>	<i>30</i>
<i>Figura G. 1. Interfaz gráfica de Dash WebM Converter.....</i>	<i>34</i>
<i>Figura G. 2. Contenidos DASH generados.....</i>	<i>35</i>
<i>Figura H. 1. Servicio de VoD</i>	<i>37</i>
<i>Figura H. 2. Servicio interactivo de chat.....</i>	<i>39</i>

Figura H. 3. Herramienta de codificación automática.....39
Figura I. 1. Partes de una prueba de usabilidad41

Anexo A

A. Generación del dataset de video

En este anexo es presentado el proceso paso a paso de la generación del dataset, en los cuales se hace uso de la API de Echonest por medio de códigos y librerías adicionales en Python. El dataset permite generar el archivo descriptor de los contenidos musicales asociados al modelo de emociones presentado en la sección 2.6.1. Este aporte permitió obtener y descargar los contenidos de video utilizados en el servicio de VoD, a través del S.R basado en contexto.

A.1 Proceso de generación dataset

A continuación se presentan los pasos seguidos para la generación del dataset de contenidos musicales de video:

1. Se declaran las librerías necesarias en Python, y se establece la llave para conexión con la API de Echonest (ver figura A.1).

```
56 #***** Librerías y Llaves *****#
57 import math,sys,youtube_dl,time,cookielib,urllib,urllib2,re,logging
58 from difflib import SequenceMatcher
59 from tinydb import TinyDB, where
60 from pyechonest import config
61 from pyechonest import artist
62 from pyechonest import track
63 from pyechonest import song
64
65 config.ECHO_NEST_API_KEY="EPX09QF26IUQIBUCP"
66 #*****#
```

Figura A. 1. Librerías utilizadas y llave de Echonest. Fuente: propia

2. Se genera la petición de canciones más populares en Echonest (ver figura A.2).

```

190 #***** Metodos Pyechonest *****#
191 def generar_script(ini):
192     while True:
193         res = song.search(sort='song_hotttness-desc', buckets=['audio_summary'], results=100, start=ini)
194
195         tam=len(res)
196         tiempos=[18,36,54,72,90]
197         lista_repetidos=list()
198
199         for i in range(tam):
200             if(i in tiempos):
201                 time.sleep(60)
202
203             cancion=res[i]
204             cur=cancion.song_currency

```

Figura A. 2. Búsqueda de canciones más populares EchoNest. Fuente: propia.

3. Se obtienen de las canciones más populares, sus características musicales mencionadas en la sección 2.3 (ver figura B.3).

```

206     if(cur not in lista_repetidos):
207         title=cancion.title
208         artist=cancion.artist_name
209         energy=cancion.audio_summary['energy']
210         valence=cancion.audio_summary['valence']
211         liveness=cancion.audio_summary['liveness']
212         tempo=cancion.audio_summary['tempo']
213         speechiness=cancion.audio_summary['speechiness']
214         acousticness=cancion.audio_summary['acousticness']
215         instrumentalness=cancion.audio_summary['instrumentalness']
216         mode=cancion.audio_summary['mode']
217         time_signature=cancion.audio_summary['time_signature']
218         duration=cancion.audio_summary['duration']
219         loudness=cancion.audio_summary['loudness']
220         danceability=cancion.audio_summary['danceability']
221         url=''
222         visua=''

```

Figura A. 3. Obtención de las características del contenido. Fuente: propia.

4. Se asocian las características de arousal y valence con un estado de ánimo, de acuerdo al modelo propuesto en la sección 2.6.1 (ver figura A.4).

```

224         e_n=2*(energy-0.5)
225         v_n=2*(valence-0.5)
226         ang=math.degrees(math.atan2(e_n,v_n))
227         if ang<0:
228             ang=ang+360
229         else:
230             ang=ang
231
232         # Estados de animo : 5 estados
233         if (ang>0 and ang<=54) or (ang>342):
234             ani2='Happy'
235         elif ang>54 and ang<=126:
236             ani2='Excited'
237         elif ang>126 and ang<=198:
238             ani2='Angry'
239         elif ang>198 and ang<=270:
240             ani2='Sad'
241         elif ang>270 and ang<=342:
242             ani2='Relaxing'

```

Figura A. 4. Obtención de la emoción del contenido. Fuente: propia.

5. Se obtiene la URL de cada contenido musical, a través de la API de Youtube; con el fin de obtener el video asociado a cada contenido musical (ver figura B.5).

```

121 def obtener_url(artist,title):
122     hdr = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11 (KHTML, like Gecko) Chrome/23.0.1271.64 Safari/537.11',
123           'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
124           'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.3',
125           'Accept-Encoding': 'none',
126           'Accept-Language': 'en-US,en;q=0.8',
127           'Connection': 'keep-alive'}
128
129     consulta=artist+' '+title+' Official Video'
130     print consulta
131     query_string = urllib.urlencode({'search_query' : consulta})
132     html_content = urllib2.Request('https://www.youtube.com/results?' + query_string, headers=hdr)
133     try:
134         response = urllib2.urlopen(html_content)
135         search_results = re.findall(r'href="/watch?v=(.+)')', response.read())
136     except urllib2.HTTPError, e:
137         print e.fd.read()
138     lista=list()
139     visitas=list()
140     datos = []
141     url_elegida=''
142     for i in range(0,len(search_results)):
143         url='http://www.youtube.com/watch?v='+search_results[i]
144         logging.getLogger().setLevel(logging.ERROR)
145         if(url not in lista):
146             try:
147                 video=obtener_info_video(url)
148                 time.sleep(2)
149                 titulo=video['title']
150                 visual=video['view_count']
151                 catego=video['categories']
152             except (EOFError, IOError, AttributeError, TypeError, ValueError, UnicodeError, UnicodeDecodeError, UnicodeEncodeError):
153                 print 'Oops! ERROR'
154                 pass
155             proba=similar(consulta,titulo)
156             print proba
157             lista.append(url)
158             if catego[0].encode('utf-8') == 'Music':
159                 if proba >= 0.6:
160                     visitas.append(visual)
161                     datos.append({'visualizaciones':visual,'url':url})
162                     print 'POSIBLE URL:',url
163                     logging.getLogger().setLevel(logging.ERROR)
164
165     maxvis=video_mvisitado(visitas)
166     for i in range(0,len(datos)):
167         if(maxvis == datos[i]['visualizaciones']):
168             url_elegida=datos[i]['url']
169     print '*****'
170     print 'URL ELEGIDA: ',url_elegida
171     print '*****'
172     datos=[url_elegida,maxvis]
173     return datos

```

Figura A. 5. Método para la obtención de url asociada. Fuente: propia.

6. Se genera un archivo JSON con la información de los contenidos, cabe mencionar que en este proyecto se seleccionaron 40 contenidos por cada uno de los cinco estados de ánimo, para un total de 200 contenidos (ver figura A.6).

```

70 #***** Metodos TinyDB *****#
71 def agregar_cancion(animo,title,artist,energy,valence,liveness,tempo,speechiness,acousticness,
72                   instrumentalness,mode,time_signature,duration,loudness,danceability,cur,url,visua):
73
74     db = TinyDB('contents.json')
75     lista=db.search((where('title')==title) & (where('artist')==artist))
76     if(len(lista)==0):
77         db.insert({'mood':animo,'title':title,'artist':artist,'energy':energy,'valence':valence,
78                 'liveness':liveness,'tempo':tempo,'speechiness':speechiness,'acousticness':acousticness,
79                 'instrumentalness':instrumentalness,'mode':mode,'time_signature':time_signature,
80                 'duration':duration,'loudness':loudness,'danceability':danceability,'currency':cur,'url':url,'view_count':visua})
81     print '*****'
82     print 'CONTENIDO ALMACENADO'
83     print '*****'
84
85 def actualizar_url_view(artist,titulo,url,visi):
86     db = TinyDB('contents.json')
87     db.update(((url:url)),((where('artist')==artista) & (where('title')==titulo)))
88     db.update(((view_count:visi)),((where('artist')==artista) & (where('title')==titulo)))
89     print '*****'
90     print 'CONTENIDO ACTUALIZADO'
91     print '*****'
92
93 def listar_codigo():
94     db = TinyDB('contents.json')
95     lista=db.all()
96     return lista
97 #*****

```

Figura A. 6. Método TinyDB y generación de archivo JSON. Fuente: propia.

7. Finalmente, son descargados los contenidos de video, a través de la librería YoutubeDL para Python (ver figura A.7).

```
175 def download_video(url, art, tit, ani):
176     nombre=ani+'/'+'art+' - '+tit+'.mp4'
177     ydl_opts = {
178         'format': '22,18',
179         'outtmpl': nombre,
180         'forcethumbnail': True,
181         'writethumbnail': True
182     }
183     with youtube_dl.YoutubeDL(ydl_opts) as ydl:
184         ydl.download([url])
185         time.sleep(2)
186         return 'Descargado: 100%'
187     #*****#
```

Figura A. 7. Descarga de contenidos multimedia. Fuente: propia.

Anexo B

B. Instalación y manejo de servidor Spark

En este anexo es presentada la instalación y creación de un servicio web básico como un HelloWorld a través del framework Spark de Java, el cual permite crear aplicaciones o servicios web sencillos y livianos en el lenguaje de programación Java. Esta herramienta es usada en el presente trabajo de grado con el fin de conformar el servidor de procesamiento junto a la herramienta de minería de datos Weka y el gestor de base de datos MySQL, encargados del módulo clasificador bayesiano el cual provee de recomendaciones al S.R. Para el presente ejemplo se hizo uso de la plataforma Eclipse Mars IDE y el navegador Google Chrome.

B.1 Creación servicio básico

A continuación, son descritos los pasos para crear un servicio web básico a través de Spark:

1. Se ejecuta Eclipse Mars y se crea un nuevo proyecto Maven, para ello deben ir a:

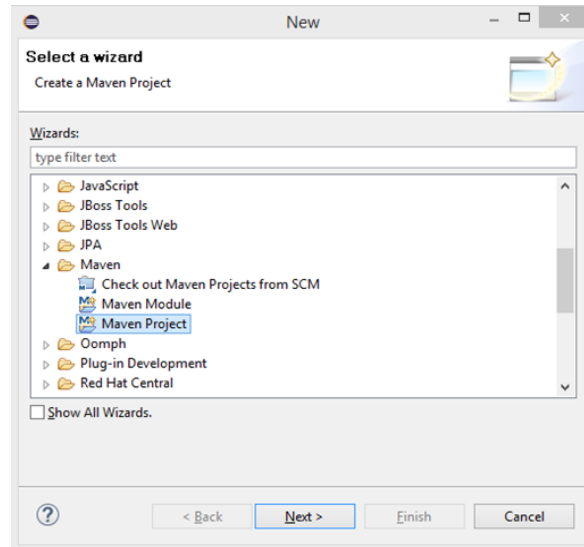
File -> New -> Other...-> Maven -> Maven Project

o

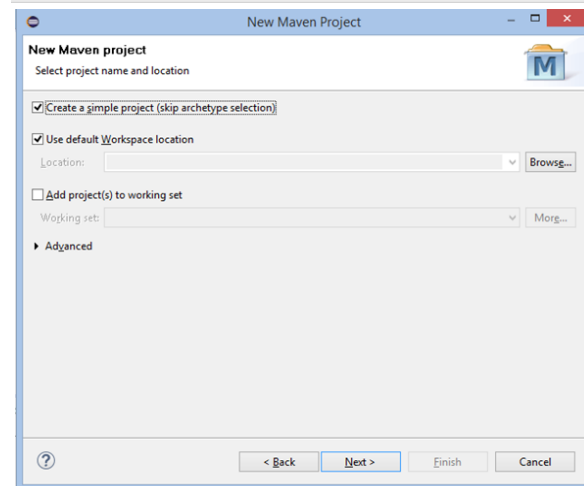
Archivo -> Nuevo -> Otros... -> Maven -> Proyecto Maven

Para ello es necesario seguir los tres siguientes pasos:

Primer paso:



Segundo paso:



Tercer paso:

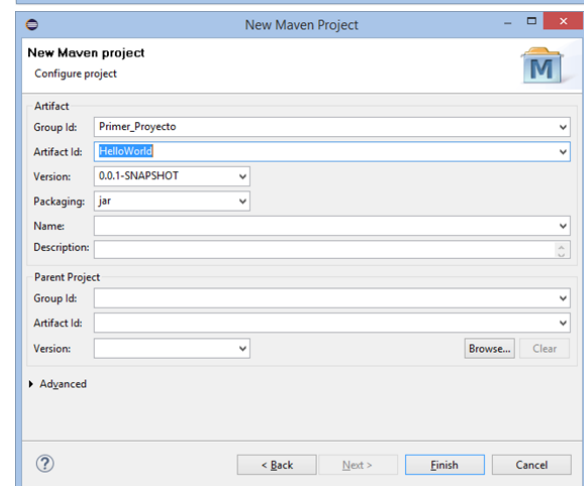
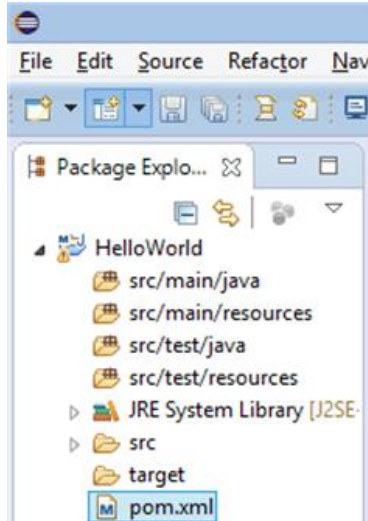


Figura B. 1. Creación proyecto maven. Fuente: propia.

2. Se agrega la dependencia de Spark en el archivo descriptor del proyecto, seguir los tres pasos siguientes:

Primer paso



Segundo paso

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/POM/4.0.0" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>Primer_Proyecto</groupId>
4   <artifactId>HelloWorld</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6
7   <dependencies>
8     <dependency>
9       <groupId>com.sparkjava</groupId>
10      <artifactId>spark-core</artifactId>
11      <version>2.5</version>
12    </dependency>
13  </dependencies>
14 </project>
```

Tercer paso

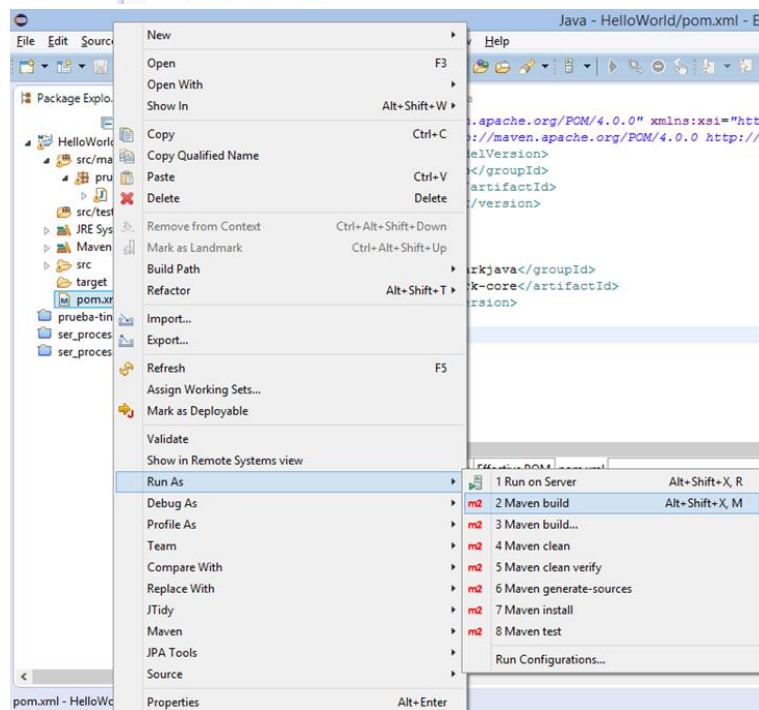
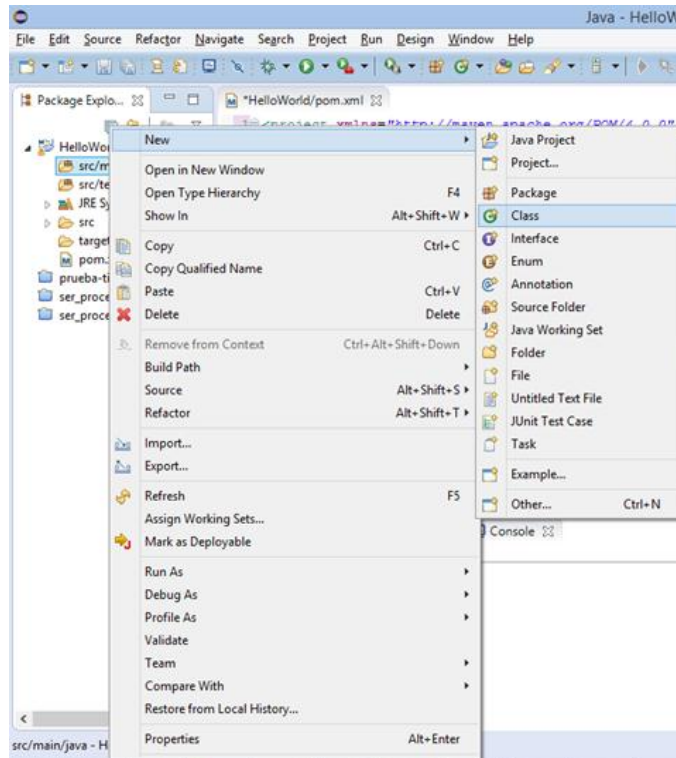


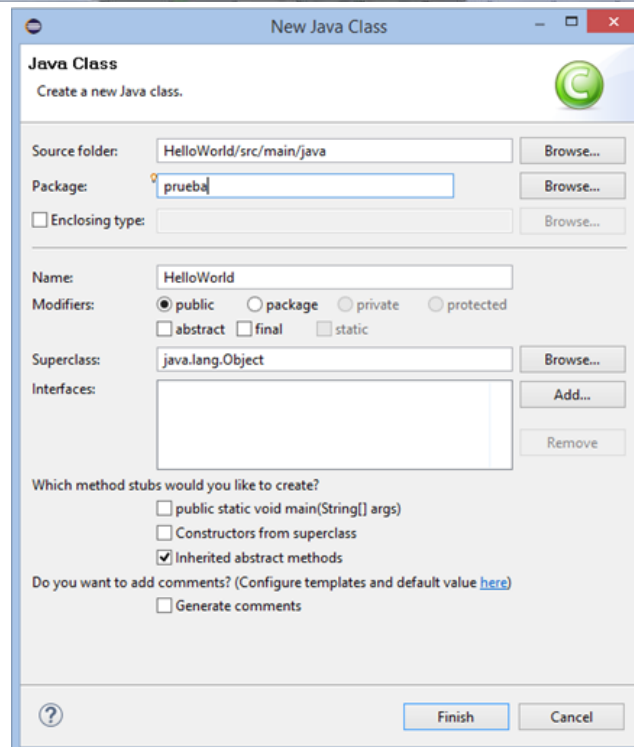
Figura B. 2. Agregar dependencia de Spark al proyecto. Fuente: propia.

3. Se crea la clase con el ejemplo de HelloWorld para el servidor Spark, ver los cuatro pasos siguientes:

Primer paso



Segundo paso



Tercer paso

```

1 package prueba;
2
3 import static spark.Spark.*;
4
5 public class HelloWorld {
6     public static void main(String[] args){
7         get("/hello", (req, res)->"<h1>HELLO WORLD</h1>");
8     }
9 }

```

Cuarto paso

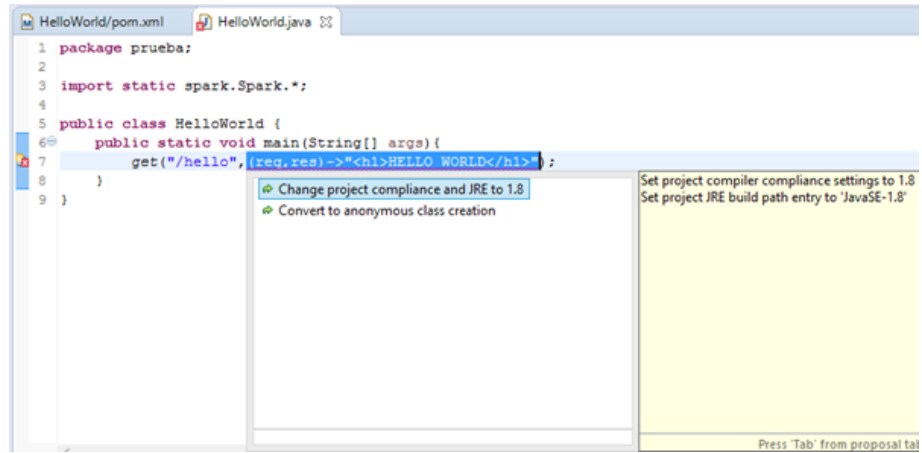
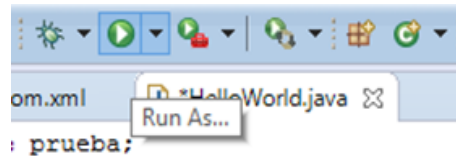


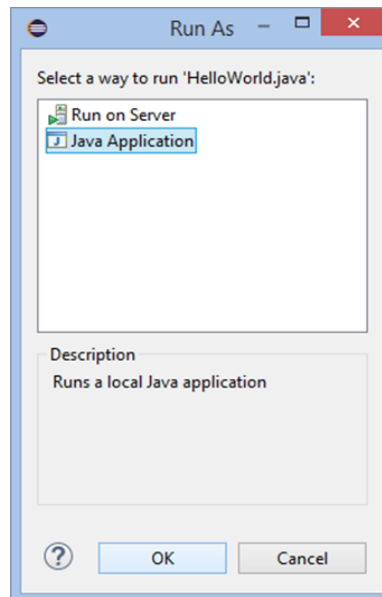
Figura B. 3. Ejemplo HelloWorld en Spark. Fuente: propia.

4. Desplegar ejemplo de HelloWorld en navegador, a través de los tres siguientes pasos:

Primer paso



Segundo paso



Tercer paso

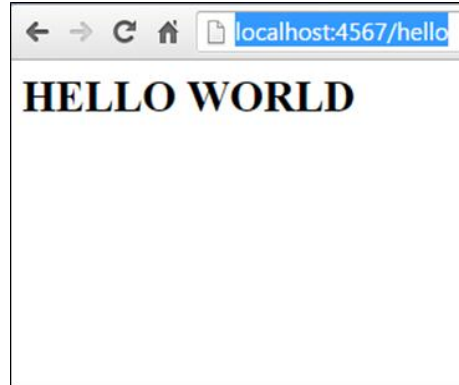


Figura B. 4. Desplegar ejemplo HelloWorld. Fuente: propia.

Anexo C

C. Instalación y manejo de servidor Flask

En este anexo es presentada la instalación y creación de un servicio web básico a través de la librería Flask de Python, el cual permite crear aplicaciones o servicios web sencillos. Esta herramienta es usada en el presente trabajo de grado para implementar el servidor de contexto, encargado del procesamiento de las variables fisiológicas y emoción de usuario, junto al módulo hardware Arduino Yún. Para el presente ejemplo se hizo uso de la herramienta Open Source Geany y el navegador Mozilla Firefox.

C.1 Instalación ejecución Flask

A continuación, son descritos los pasos para la instalación del framework Flask en el sistema operativo Linux:

1. Se instala el lenguaje de programación Python, en caso de que su versión de Linux ya brinde soporte de este lenguaje saltar este paso (ver figura C.1).

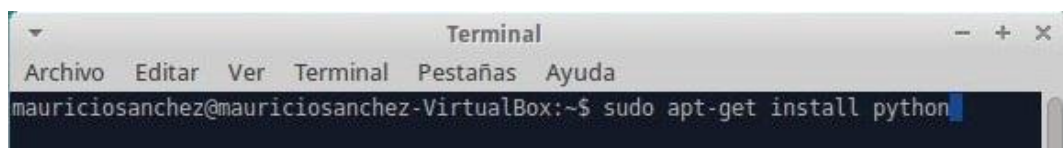
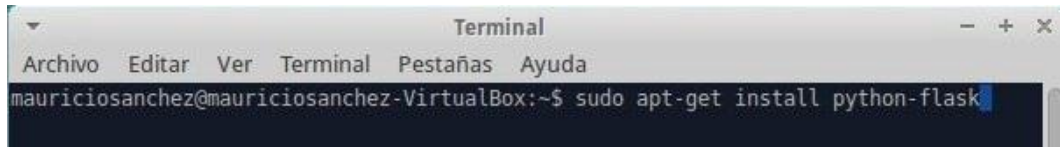


Figura C. 1. Instalación de Python en Linux. Fuente: propia.

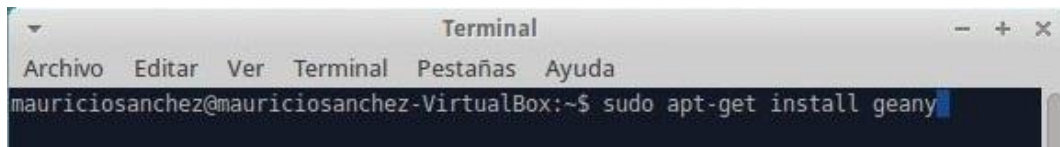
2. Se instala la librería de Flask para Python (ver figura C.2).



```
Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
mauriciosanchez@mauriciosanchez-VirtualBox:~$ sudo apt-get install python-flask
```

Figura C. 2. Instalación de librería Flask. Fuente: propia.

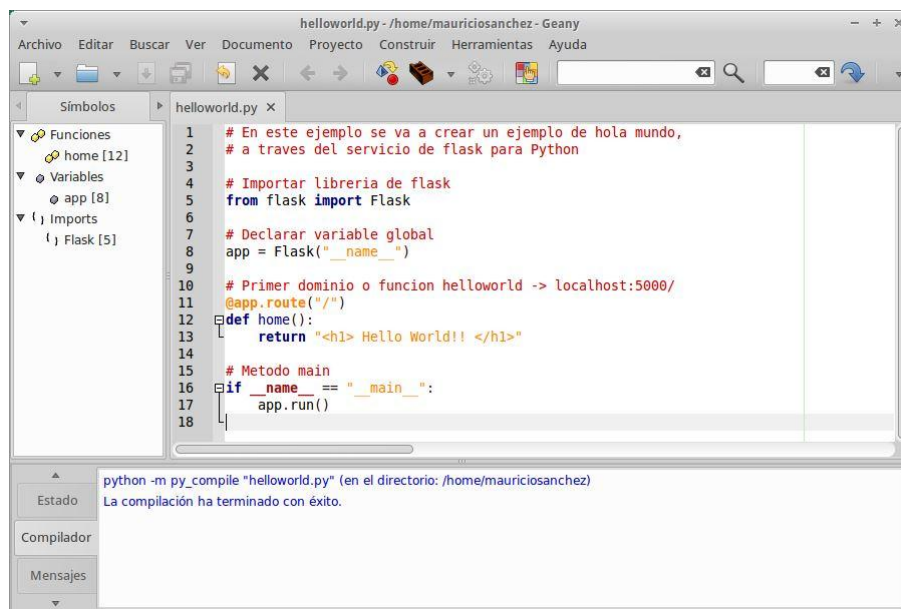
3. Se instala el entorno de desarrollo Geany, para la crear proyecto HelloWorld en Flask (ver figura C.3).



```
Terminal
Archivo Editar Ver Terminal Pestañas Ayuda
mauriciosanchez@mauriciosanchez-VirtualBox:~$ sudo apt-get install geany
```

Figura C. 3. Instalación del entorno de desarrollo Geany. Fuente: propia.

4. Se abre el entorno de desarrollo Geany, desde la consola terminal con el comando “geany” y se crea el *script* para el helloworld.py de Python (ver figura C.4).

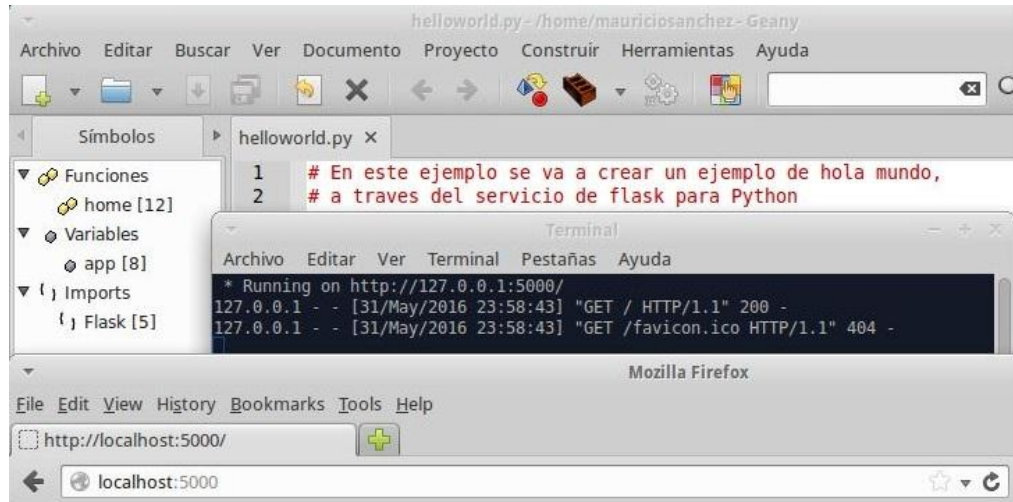


```
helloworld.py - /home/mauriciosanchez - Geany
Archivo Editar Buscar Ver Documento Proyecto Construir Herramientas Ayuda
Simbolos helloworld.py x
Funciones
  home [12]
Variables
  app [8]
Imports
  Flask [5]
1 # En este ejemplo se va a crear un ejemplo de hola mundo,
2 # a través del servicio de flask para Python
3
4 # Importar librería de flask
5 from flask import Flask
6
7 # Declarar variable global
8 app = Flask("__name__")
9
10 # Primer dominio o función helloworld -> localhost:5000/
11 @app.route("/")
12 def home():
13     return "<h1> Hello World!! </h1>"
14
15 # Metodo main
16 if __name__ == "__main__":
17     app.run()
18
Estado python -m py_compile "helloworld.py" (en el directorio: /home/mauriciosanchez)
La compilación ha terminado con éxito.
Compilador
Mensajes
```

Figura C. 4. Proyecto HelloWorld en Flask. Fuente: propia

Como es observado en la figura C.4, primero es importada la librería de flask para Python, luego es necesario crear una variable global estática con la cual se va arrancar el proyecto. Luego es creado el dominio “/” con la función home(), la cual va a retornar el mensaje “Hello World!!” cada vez que es invocada la petición: localhost:5000/. Y finalmente es declarado el método main, el cual se encarga de arrancar el proyecto.

5. Se lanza el servidor Flask a través del botón compilar y es desplegado en el navegador Mozilla Firefox con la petición http://localhost:5000/.



Hello World!!

Figura C. 5. Ejecución del proyecto HelloWorld en Flask. Fuente: propia.

Anexo D

D. Manejo plataforma de Arduino Yún

En este anexo es presentada la configuración, creación y despliegue de un servicio web soportado mediante la plataforma Arduino Yún.

D.1 Servicio web con Arduino Yún

En primera instancia es necesario para iniciar la configuración del dispositivo, ingresar a su interfaz de red por medio del módulo wifi o ethernet, para motivos del ejemplo se hace uso de la interfaz de red wifi, como es apreciado en la figura D.1.

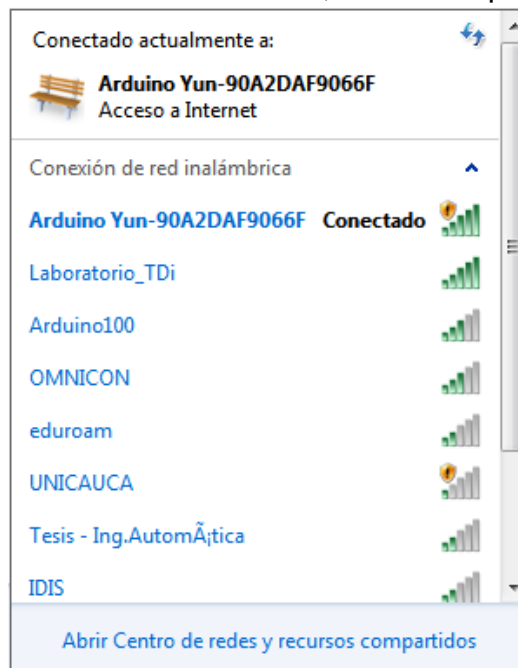


Figura D. 1. Conexión a la red arduino yún. Fuente: propia.

Luego de estar conectado al Arduino Yún, en el navegador web del computador se ingresa a la puerta de enlace de red con la ip 192.168.240.1, esta pide una contraseña para permitir el acceso a la configuración que por defecto será “arduino”, una vez se realiza la validación es presentada la siguiente interfaz (ver figura D.2).

YÚN BOARD CONFIGURATION

YÚN NAME * servidorcontexto

PASSWORD

CONFIRM PASSWORD

TIMEZONE * America/Bogota

WIRELESS PARAMETERS

CONFIGURE A WIRELESS NETWORK

DETECTED WIRELESS NETWORKS Laboratorio_TDi (WPA2) Refresh

WIRELESS NAME * Laboratorio_TDi

SECURITY WPA2

PASSWORD *

DISCARD CONFIGURE & RESTART

REST API ACCESS

REST API ACCESS OPEN WITH PASSWORD

REST APIs allow you to access your sketch from the web, sending commands or exchanging configuration values.
If your Yún is on a public network, or controlling sensitive equipment, or both, we recommend you leave the REST API password protected.

Figura D. 2. Interfaz de configuración arduino yún. Fuente: propia.

En la interfaz de la figura D.2 se tienen 3 aspectos a configurar, como son la tarjeta yún y los parámetros de la red, y el acceso a la API rest. En la primera es ingresada la información referente a la validación para poder realizar modificación en la tarjeta y la carga de códigos. La segunda hace parte de la configuración de la red con la cual se piensa conectar la tarjeta Arduino Yún. Por último, es posible seleccionar si para el acceso al servicio es necesario ingresar una contraseña o por lo contrario cualquiera puede tener acceso y hacer uso del servicio.

Una vez es presionado el botón configurar y resetear, es guardada toda la información y es posible hacer uso de la tarjeta. Para ello en el IDE de Arduino en herramientas es escogida la placa y el puerto que se reconocerá con el nombre

configurado en la yún seguido de la ip, luego de esto se procede con la carga del código en el dispositivo (ver figura D.3)

```
holamundo$  
// se declaran la librerías para el montaje del servidor y el cliente HTTP  
#include <Bridge.h>  
#include <BridgeServer.h>  
#include <BridgeClient.h>  
BridgeServer server;  
  
void setup() {  
  Serial.begin(115200);  
  Bridge.begin();  
  // se indica al servidor para empezar a escuchar las conexiones entrantes puerto:5555  
  server.listenOnLocalhost();  
  server.begin();  
}  
  
void loop() {  
  // se crea el objeto que hace uso de la clase BridgeClient  
  BridgeClient cliente = server.accept();  
  if(cliente){  
    String comando = cliente.readString();  
    comando.trim();  
    Serial.println(comando);  
    if(comando == "helloworld"){  
      cliente.print("Hello World!!");  
    }  
    cliente.stop();  
  }  
  delay(50);  
}
```

Figura D. 3. Figura D.3. Código prueba de arduino yún. Fuente: propia.

Para acceder a la API rest de Arduino, se ingresa en el navegador la url con la petición al servidor, esta contiene: los datos de ip de la Arduino, una marca con la representación “arduino” y el comando para lanzar la aplicación, en este caso es “helloworld” (ver figura D.4).

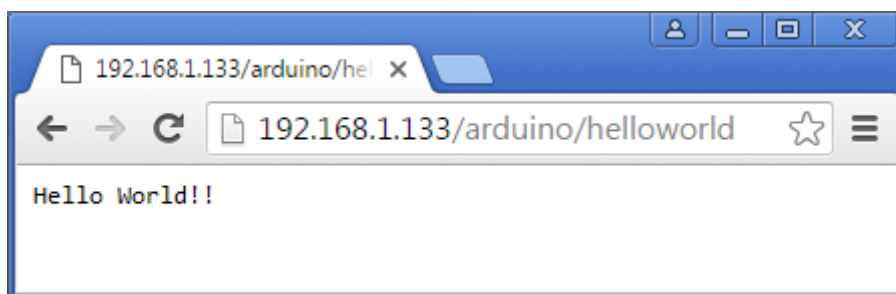


Figura D. 4. Ejemplo de Hola mundo a través de la tarjeta arduino yún. Fuente: propia.

Anexo E

E. Herramienta Weka 3.6

En este anexo es mostrado el uso básico de la herramienta Weka 3.6, la cual contiene una serie de algoritmos de aprendizaje automático para tareas de minería de datos, de dónde se implementa el algoritmo de Naive Bayes para el desarrollo del clasificador del presente trabajo de grado. Para ello se hará uso de la interfaz gráfica con la cual esta cuenta, además, este proceso puede ser llevado a cabo por medio de la consola de comandos o través de la API provista por la herramienta. La herramienta Weka trabaja con distintas fuentes de datos como son documentos, URL y base de datos de distintos formatos.

E.1 Clasificador bayesiano con Weka 3.6

Para ilustrar el proceso de creación de clasificador bayesiano, la fuente de entrada de los datos en este caso es un documento de formato arff (Attribute-Relation file format) y es con el cual trabaja Weka, el cual presenta la siguiente estructura (ver figura E.1).

```
@relation peliculas

@attribute genero {drama,comedia, accion}
@attribute ano {2013,2014,2015}
@attribute class {buena,regular,mala}

@data
drama,2013,buena
drama,2013,regular
comedia,2013,mala
comedia,2014,regular
accion,2013,mala
comedia,2013,mala
comedia,2014,regular
drama,2013,buena
drama,2015,regular
drama,2015,mala
drama,2014,?
```

Figura E. 1. Archivo arff weka. Fuente: propia.

El documento arff es cargado en el explorador de Weka de la figura E.2, y el cual contiene la información de un conjunto de películas, esto con el fin de solicitar la predicción del atributo class que tendría una película de género drama del 2014.

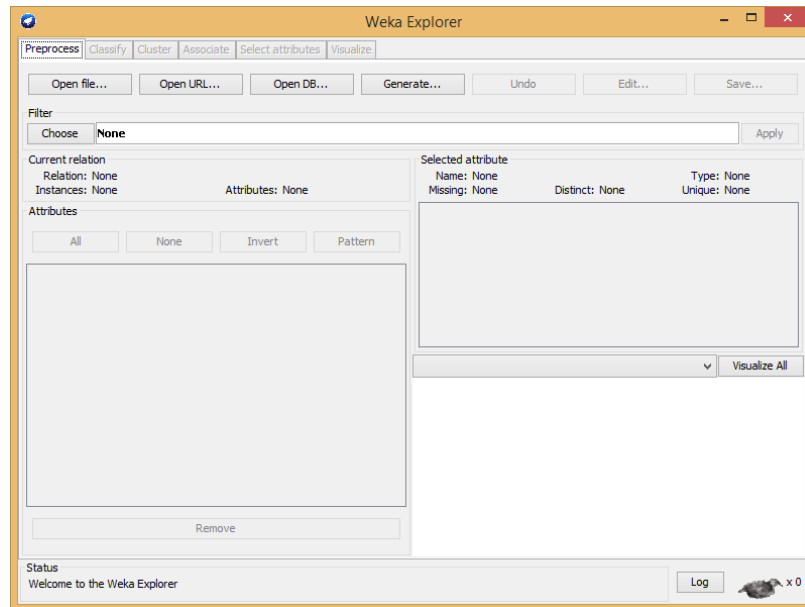


Figura E. 2. Explorador herramienta weka. Fuente: propia.

En el explorador es cargado el archivo, luego la herramienta permite visualizar de forma gráfica los atributos del documento como es observado en la figura E.3.

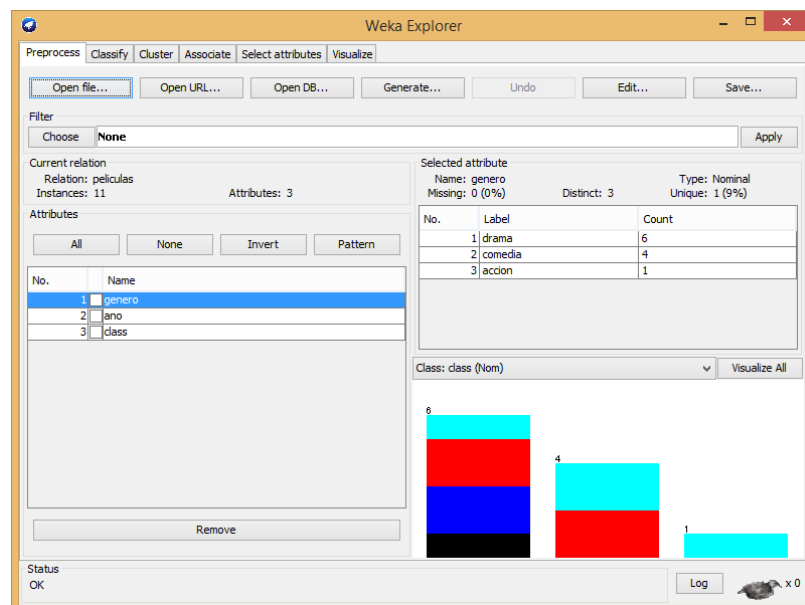


Figura E. 3. Explorador herramienta weka. Fuente: propia.

La herramienta Weka permite seleccionar entre distintos tipos de filtros con el fin de realizar transformaciones sobre los atributos del documento (ver figura E.4), en caso de querer hacer un tratamiento especial para los datos de estudio, para lo cual

solo se debe pulsar el botón de choose y elegir uno. Si no es el caso bien es posible empezar la clasificación de estos.

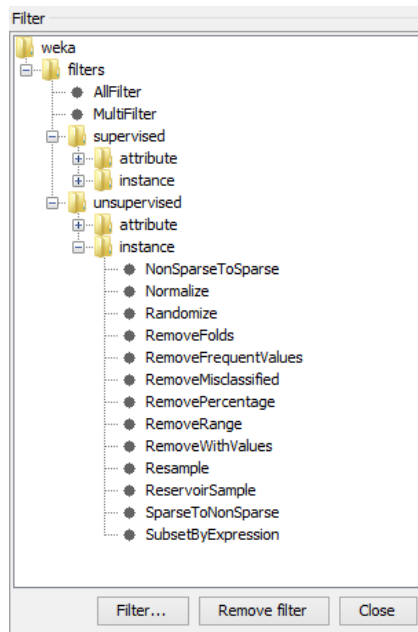


Figura E. 4. Filtros de weka. Fuente: propia.

Una vez cargados los datos en la pestaña de Classify o clasificar se encuentran los distintos métodos que se pueden aplicar para el estudio de los datos como es mostrado en la figura E.5. Además, al pulsar sobre el clasificador escogido es posible configurar las propiedades de este según la necesidad. En este caso se hará uso del clasificador basado en Naive Bayes para el análisis de los datos del archivo arff.

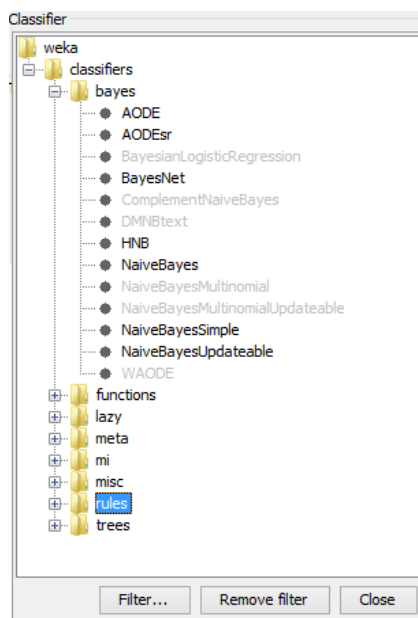


Figura E. 5. Algoritmos de clasificación weka. Fuente: propia.

Al aplicar el tipo de clasificador y escoger las opciones del test en este caso fue el (use training set), es decir, la herramienta entrenara el método elegido con los datos proporcionados y luego lo aplicará otra vez sobre los mismos. Para ello se da inicio a la herramienta y en la pantalla del explorador de Weka es desplegada toda la información de análisis que provee la misma con relación a los datos proporcionados, como es apreciado en la figura E.7. Una vez se haya habilitado la pestaña (Output predictions) en las opciones de evaluación del clasificador (ver figura E.6) la cual permite el despliegue en pantalla del análisis, donde la herramienta proporciona las predicciones de los atributos.

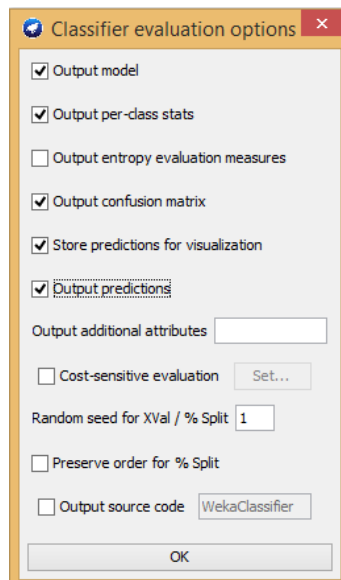


Figura E. 6. Opciones de salida. Fuente: propia

Classifier output

```

=== Predictions on training set ===
inst#,  actual, predicted, error, probability distribution
1  1:buena  1:buena      *0.431 0.244 0.325
2  2:regular 1:buena      + *0.431 0.244 0.325
3  3:mala   3:mala       0.164 0.279 *0.557
4  2:regular 2:regular    0.089 *0.683 0.228
5  3:mala   3:mala       0.261 0.148 *0.591
6  3:mala   3:mala       0.164 0.279 *0.557
7  2:regular 2:regular    0.089 *0.683 0.228
8  1:buena  1:buena      *0.431 0.244 0.325
9  2:regular 2:regular    0.261 *0.444 0.296
10 3:mala   2:regular    + 0.261 *0.444 0.296
11 ? 2:regular  + 0.243 *0.619 0.138

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      8          80  %
Incorrectly Classified Instances    2          20  %
Kappa statistic                    0.697
Mean absolute error                 0.3389
Root mean squared error             0.3762

```

Figura E. 7. Análisis de la herramienta weka sobre los datos iniciales. Fuente: propia

La herramienta Weka proporciona información sobre las distintas predicciones realizadas, resumen de métricas estadísticas en cuanto el error como lo son: Root mean squared error, Mean absolute error, Relative absolute error, entre otros, además proporciona información relevante en cuenta el error del clasificador conocida como matriz de confusión, en este caso la predicción del atributo class realizada por la aplicación para la película drama del año 2014 es "regular", con una probabilidad de 0.619.

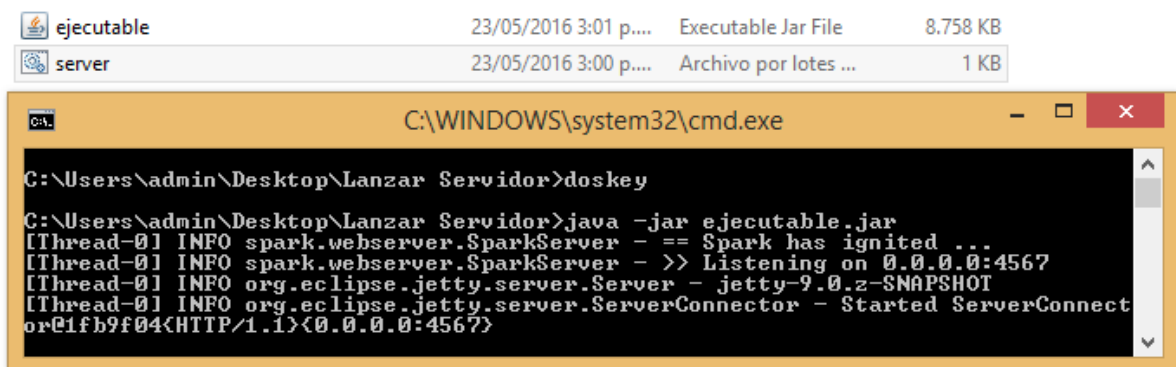
Anexo F

F. Herramienta apache benchmark

A continuación, es descrito el uso de la herramienta apache benchmark, la cual fue utilizada en el presente trabajo para la realización de pruebas de carga y estrés sobre el servidor de procesamiento y el servicio web proporcionado por módulo Arduino para la obtención de las variables fisiológicas. Para este ejemplo se hará uso del paquete software XAMP, el cual contiene en la carpeta de instalación el servidor apache con la herramienta ab.exe la cual es ejecutada desde CMD.

F.1 Manejo apache benchmark

Primero antes de usar la herramienta se debe verificar si el servidor que aloja la aplicación está activo, para motivos de este ejemplo se ha puesto en funcionamiento el servidor de procesamiento trabajado en el proyecto como es apreciado en la figura F.1.

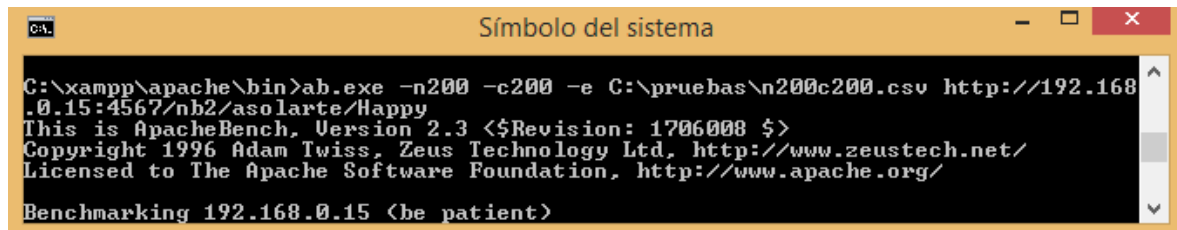


The image shows a Windows Explorer window with two files: 'ejecutable' (8.758 KB, Executable Jar File) and 'server' (1 KB, Archivo por lotes ...). Below it is a Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe' showing the following output:

```
C:\Users\admin\Desktop\Lanzar Servidor>doskey
C:\Users\admin\Desktop\Lanzar Servidor>java -jar ejecutable.jar
[Thread-0] INFO spark.webserver.SparkServer - == Spark has ignited ...
[Thread-0] INFO spark.webserver.SparkServer - >> Listening on 0.0.0.0:4567
[Thread-0] INFO org.eclipse.jetty.server.Server - jetty-9.0.z-SNAPSHOT
[Thread-0] INFO org.eclipse.jetty.server.ServerConnector - Started ServerConnector@1fbb9f04{HTTP/1.1}{0.0.0.0:4567}
```

Figura F. 1. Lanzando el servidor. Fuente: propia

Luego es ejecutado el comando para ello si la carpeta del programa no se encuentra en el path del sistema, es necesario ingresar, en la carpeta bin de apache e ingresar el comando de la figura F.2.



```

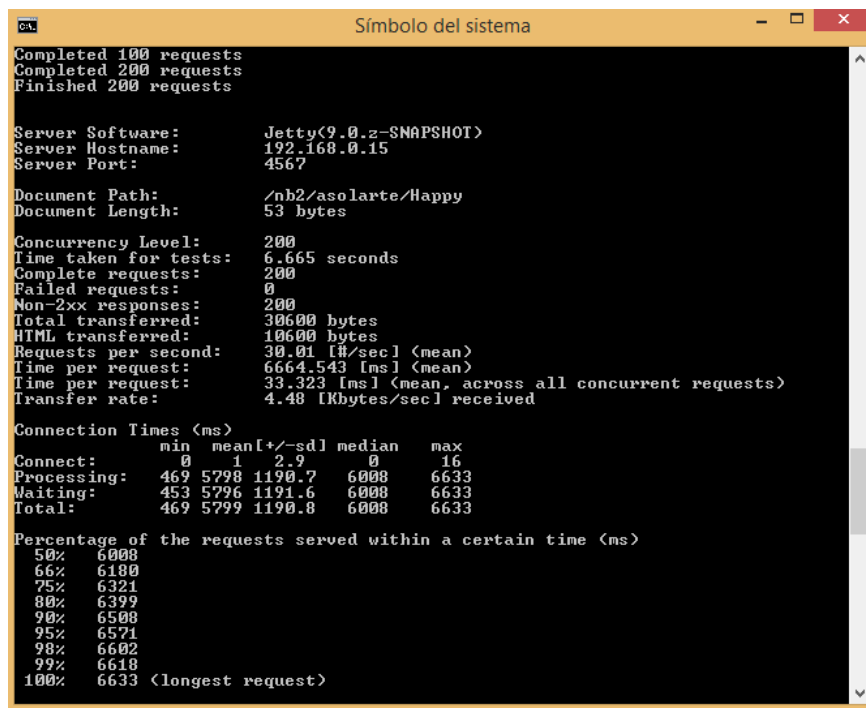
C:\xampp\apache\bin>ab.exe -n200 -c200 -e C:\pruebas\n200c200.csv http://192.168
.0.15:4567/nb2/asolarte/Happy
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.0.15 <be patient>

```

Figura F. 2. Comando apache-benchmark. Fuente: propia.

Donde ab hace referencia al apache benchmark, para el número de conexiones a realizar es empleado el comando `-n`, para determinar el número de conexiones concurrentes que se deseen hacer se utiliza `-c`, y si se quiere guardar un archivo con el tiempo máximo en dar respuesta el servidor se hace uso del comando `-e`, lo cual genera un documento "csv". Después de haber establecido los parámetros se ingresa la url y se ejecuta y en la misma consola es obtenido un resumen con los resultados como es apreciado en la figura F.3.



```

Completed 100 requests
Completed 200 requests
Finished 200 requests

Server Software:      Jetty(9.0.z-SNAPSHOT)
Server Hostname:     192.168.0.15
Server Port:         4567

Document Path:       /nb2/asolarte/Happy
Document Length:     53 bytes

Concurrency Level:   200
Time taken for tests: 6.665 seconds
Complete requests:   200
Failed requests:     0
Non-2xx responses:   200
Total transferred:   30600 bytes
HTML transferred:   10600 bytes
Requests per second: 30.01 [#/sec] (mean)
Time per request:    6664.543 [ms] (mean)
Time per request:    33.323 [ms] (mean, across all concurrent requests)
Transfer rate:       4.48 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    1    2.9      0   16
Processing: 469  5798 1190.7    6008 6633
Waiting:    453  5796 1191.6    6008 6633
Total:      469  5799 1190.8    6008 6633

Percentage of the requests served within a certain time (ms)
 50%: 6008
 66%: 6180
 75%: 6321
 80%: 6399
 90%: 6508
 95%: 6571
 98%: 6602
 99%: 6618
100%: 6633 (longest request)

```

Figura F. 3. Despliegue de resultados. Fuente: propia.

Donde la información relevante de los resultados es: Requests per second o conexiones atendidas por segundo durante la prueba, time per request (mean) o tiempo medio donde el servidor ha tardado en atender un grupo de conexiones concurrentes y time per request (mean, across all concurrent request) o tiempo medio donde el servidor tardó en atender una petición de forma individual.

Anexo G

G. Generación de contenidos DASH-WebM

Las siguientes herramientas han sido seleccionadas en este trabajo, para el proceso de generación de contenidos multimedia DASH y están indicadas según su orden de participación durante el proceso:

- FFmpeg
- Libwebm
- Webm-tools

Estas herramientas deben haber sido compiladas e instaladas con anterioridad en un sistema operativo Linux, del mismo modo, los archivos multimedia a codificar deberán tener el formato WebM y estar dispuestos en un directorio de fácil acceso.

G1. Generación flujos de video y audio

Primero desde una terminal de comandos Linux, se debe navegar hasta el directorio donde están almacenados los archivos multimedia en formato WebM, por ejemplo:

```
cd /Documentos/videos/
```

Una vez ubicados en la carpeta anterior, se procede a la codificación de los contenidos multimedia a diferentes tasas de bit. Para ejecutar correctamente la herramienta de codificación FFmpeg se debe pasar como parámetros:

- ✓ El nombre del archivo multimedia a codificar, incluyendo la extensión del mismo (.webm).
- ✓ El códec de video adecuado.

- ✓ La tasa de bit a la cual se desea codificar el flujo de video.
- ✓ Un tamaño de resolución de pantalla.
- ✓ El parámetro para indicar que solamente se desea un flujo de video.
- ✓ El nombre del flujo de video de salida con la extensión del formato (.webm).

Las siguientes líneas describen los comandos correctos para realizar la codificación de flujos de video a diferentes tasas de bit:

```
ffmpeg -i archivo.webm -an -vcodec libvpx -vf scale:320:240 -vb 100k  
-keyint_min 150 -g 150 stream_video1.webm
```

```
ffmpeg -i archivo.webm -an -vcodec libvpx -vf scale:320:240 -vb 200k  
-keyint_min 150 -g 150 stream_video2.webm
```

De igual modo para generar un flujo de audio debe digitarse la siguiente instrucción:

```
ffmpeg -i archivo.webm -vn -acodec libvorbis -ab 128k stream_audio.webm
```

En este punto ya están listos los flujos de video y el flujo de audio para continuar con la siguiente parte de la generación de contenidos multimedia DASH.

G2. Indexación de contenidos

A partir de los contenidos codificados en el paso anterior, en este paso se añaden índices de búsqueda a cada flujo de video, es decir, los fotogramas clave para realizar el canjeo entre los *stream* durante la reproducción del contenido multimedia DASH.

Desde una terminal de comandos de Linux y ubicados en el directorio donde están alojados los flujos de video y audio generados en la anterior parte, debe ejecutarse el *script sample_muxer* de la herramienta Libwebm. La instrucción debe contener como parámetros el nombre del archivo origen y el nombre del archivo destino con la extensión del formato (.webm), como se indica a continuación:

```
./libwebm/samplemuxer -i stream_video1.webm -o  
stream_video1_sm.webm
```

```
./libwebm/samplemuxer -i stream_video2.webm -o  
stream_video2_sm.webm
```

Para el flujo de audio de esta otra manera:

```
./libwebm/samplemuxer -i stream_audio.webm -o stream_audio_sm.webm
-output_cues 1 -cues_on_audio_track 1 -max_cluster_duration 5
-audio_track_number 2
```

Puesto que ya no es necesario conservar los anteriores flujos de video y audio, se procede a su eliminación manualmente o desde la misma terminal, digitando:

```
rm -r stream_video1.webm
rm -r stream_video2.webm
rm -r stream_audio.webm
```

Los nuevos flujos de video y audio ahora están listos para generar el archivo de manifiesto MPD, por lo tanto, en la siguiente parte se describirá la forma de hacerlo.

G3. Generación del archivo MPD

Finalmente para cumplir con la especificación del estándar DASH, se debe generar un archivo descriptor MPD, por lo tanto es necesario ejecutar el *script webm_dash_manifest* de la herramienta Webm-tools, para ello desde una terminal de Linux se incluyen como parámetros cada uno de los flujos de video y audio generados en la anterior sección, la instrucción sería algo semejante a la siguiente línea:

```
./webm-tools/webm_dash_manifest -o archivo_manifiesto.mpd
-as id=0,leng=eng -r id=0,file=stream_video1_sm.webm
-r id=1,file=stream_video2_sm.webm
...
-r id=N-1,file=stream_videoN_sm.webm
-as id=1,leng=eng -r id=N,file=stream_audio_sm.webm
```

La expresión **N-1** y **N** indican la secuencia del parámetro *id* según la cantidad de flujos de video que hayan sido codificados.

En este punto el proceso de generación de contenidos multimedia DASH, ha terminado de manera satisfactoria. Sin embargo, al momento de tratar de adaptar nuevos contenidos WebM al estándar, es necesario volver a iniciar el mismo proceso desde la sección B1. Por esta razón la siguiente sección explicará el mismo proceso con ayuda de la herramienta de codificación automática, implementada en este trabajo.

G4. Proceso de codificación automática

A partir de las anteriores secciones de este anexo, puede observarse cada una de las tareas específicas que se deben realizar para generar contenidos multimedia DASH. Por lo tanto en esta sección se presenta como alternativa el uso de la herramienta de codificación automática, desarrollada en este trabajo.

La herramienta Dash WebM Converter, se encargará de realizar el anterior proceso de forma automática, donde el usuario solo tendrá que seleccionar un par de parámetros para iniciar la codificación. A continuación en la figura G.4 se muestra la interfaz gráfica de dicha herramienta.

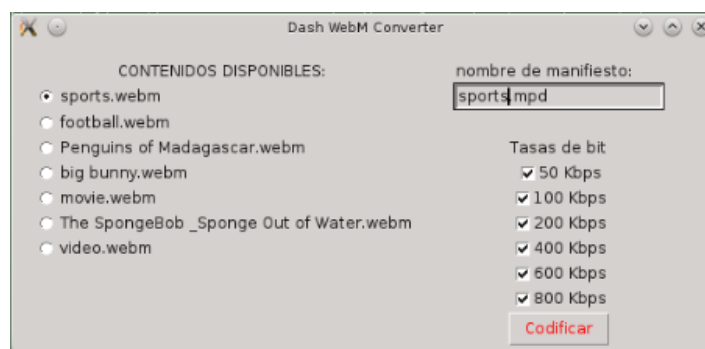


Figura G. 1. Interfaz gráfica de Dash WebM Converter. Fuente: propia.

La función de la herramienta Dash WebM Converter es ejecutar de manera secuencial las tareas de codificación (generación de flujos de video y audio), segmentación (indexación de contenidos) y generación del archivo de manifiesto MPD para contenidos multimedia DASH en formato WebM. A continuación se enumeran cada una de las funciones específicas ejecutadas con ayuda de la herramienta:

1. Se debe seleccionar el archivo multimedia WebM a codificar, mediante el sistema de selección de la herramienta de codificación automática.
2. Es necesario seleccionar las tasas de bit para la generación de los flujos de video.
3. También se debe indicar un nombre para el archivo de manifiesto MPD.
4. Finalmente para dar inicio al proceso de codificación automática, se presiona el botón con la etiqueta codificar.

Automáticamente la herramienta sigue la siguiente secuencia de tareas:

- Se crea el directorio que va a contener los flujos de video y audio junto a su respectivo archivo descriptor MPD.
- Se generan los flujos de video según las tasas de bit seleccionadas y el flujo de audio a una tasa fija de 128 Kbps.
- Se agregan los índices de búsqueda a cada uno de los flujos de video y audio generados.
- Se eliminan los flujos de video y audio que conformaron el proceso de pre codificación.

Finalmente en la figura b.4.2 se muestra el directorio generado por la herramienta, en la cual se pueden observar los flujos de video y audio adaptados al estándar junto a su respectivo archivo descriptor MPD.

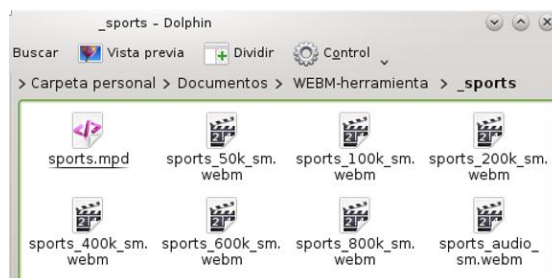


Figura G. 2. Contenidos DASH generados. Fuente: propia.

Anexo H

H. Servicio de VoD DASH

A continuación se describen cada una de las funcionalidades del servicio de VoD DASH, dispuestas para los usuarios que accedan al mismo.

F.1 Manual de usuario

Primero se debe acceder al servicio ingresando la URL de su ubicación desde un navegador Chrome en su versión 17 o superior a esta. La figura H.1 muestra el servicio de VoD en funcionamiento.



Figura H. 1. Servicio de VoD. Fuente: propia.

Los contenidos presentados al usuario corresponden a los Tráilers de las películas más recientes en el cine. Las numeraciones presentes en la figura D.1 corresponden a las siguientes funcionalidades o características:

1. **Etiqueta de título:** Corresponde a la etiqueta que muestra el nombre de la película que actualmente se está reproduciendo.
2. **Reproductor de contenidos:** Es el marco en donde se podrá visualizar la película previamente elegida por el usuario. Este reproductor está basado en el uso de HTML5.
3. **Media:** Corresponde a la pestaña que despliega el catálogo de contenidos multimedia del servicio de VoD ofrecido al usuario.
4. **Selector de género:** Es una lista desplegable desde la cual el usuario elige el tipo de género de película de su preferencia. Una vez seleccionado el género se actualiza el listado de películas mostrado en la parte inferior del selector de género.
5. **Poster:** Es la captura de una escena de la película. Al dar clic sobre esta se inicia el despliegue de la película en el reproductor de contenidos.
6. **Información del contenido:** Consiste en la información más relevante del contenido disponible para el usuario, entre ella está el género de película al que pertenece dicho contenido, el año de lanzamiento de la película y el número de veces que ha sido reproducido o visitado.
7. **URL de descriptor:** Corresponde a una etiqueta que contiene la ubicación del archivo descriptor MPD, este enlace puede ser utilizado por otro cliente web basado en el estándar DASH, para hacer uso de los contenidos dispuestos en el servicio de VoD.
8. **Sistema de valoración:** Es un servicio interactivo para la valoración de contenidos multimedia del servicio de VoD, el cual permite ordenar las películas teniendo en cuenta las valoraciones hechas por los usuarios.
9. **Chat:** Es una pestaña para acceder al servicio interactivo de chat asociado al servicio de VoD. Permite al usuario participar dentro del chat general de la comunidad en la cual se encuentra registrado, ver Figura H.2.

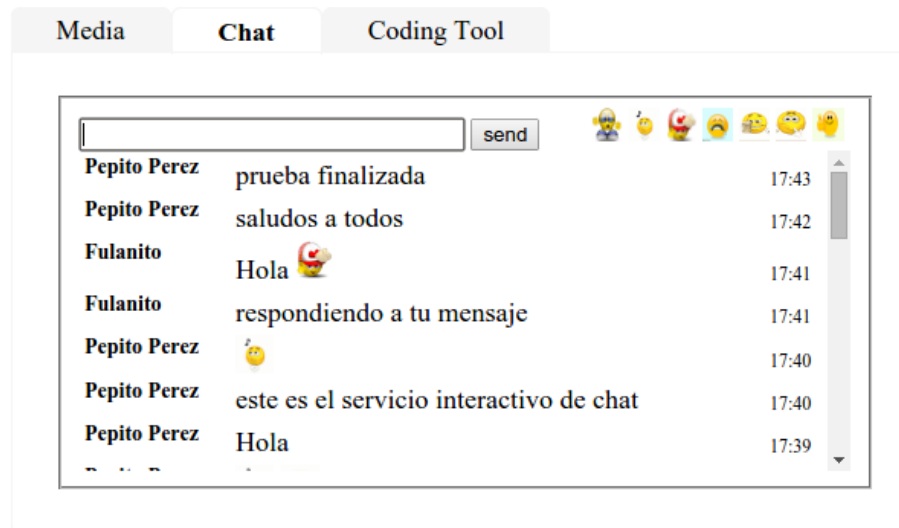


Figura H. 2. Servicio interactivo de chat. Fuente: propia.

10. **Herramienta de codificación automática:** Es una pestaña para acceder a un componente administrativo del servicio de VoD. Desde esta es posible generar contenidos multimedia DASH y publicarlos en el catálogo de contenidos del servicio, ver Figura D.3.



Figura H. 3. Herramienta de codificación automática. Fuente: propia.

11. **Medidor de ancho de banda:** Es un sistema para visualizar la calidad de imagen del video que actualmente se está desplegando.
12. **Botón de medidor de ancho de banda:** Corresponde a un botón para visualizar u ocultar la gráfica el medidor de ancho de banda.

Anexo I

I. Prueba de usabilidad películashoy.com

A continuación se ilustran las diferentes partes del instrumento generado para la prueba de usabilidad sobre el servicio de películashoy.com. A partir de esta prueba se obtuvo una caracterización del servicio de VoD.

I1. Instrumento de usabilidad

El instrumento usado para la prueba consta de: un acuerdo de confidencialidad, un cuestionario pre-test, listado de tareas y un cuestionario pos-test (ver Figura I1). En el acuerdo de confidencialidad se garantiza que los datos obtenidos serán usados solo con propósitos académicos. El cuestionario pre-test a su vez tiene por objetivo capturar un perfil inicial del usuario, así como su experiencia en el uso de portales relacionados con el portal de películashoy.com. El listado de tareas recopila un conjunto de tareas puntuales con respecto al manejo del sitio películashoy.com. Finalmente el cuestionario pos-test tiene por objetivo capturar la percepción del usuario con respecto al manejo del sitio de manera cualitativa y cuantitativa. A continuación se presenta el instrumento generado para la prueba de usabilidad sobre el sitio películashoy.com.

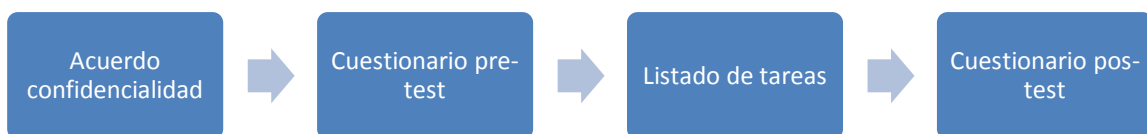


Figura I. 1. Partes de una prueba de usabilidad. Fuente: propia.

Acuerdo de confidencialidad

YO, _____ ACEPTO participar en una prueba de usabilidad supervisada por Gabriel Elías Chanchí Golondrino, el día 13/05/2014, en el *Laboratorio de Usabilidad* de la *Escuela de Ingeniería Informática* de la *Pontificia Universidad Católica de Valparaíso*. Entiendo y estoy de acuerdo con las condiciones mencionadas en adelante.

Entiendo que el experimento tiene por objetivo evaluar un sistema software, NO mis capacidades/habilidades/conocimientos.

Entiendo que los resultados del experimento se utilizaran sólo para propósitos académicos y/o de investigación, sin que mi identidad sea revelada.

Entiendo que puedo comunicar al supervisor del experimento, en cualquier momento, sobre algún malestar, molestia o inconformidad que pueda sentir durante el desarrollo del experimento; y que por tal motivo, puedo abandonar el experimento y el laboratorio en cualquier momento.

Firma

Estimado(a) colaborador(a):

Usted participará en una prueba para evaluar el grado de usabilidad del portal de **peliculashoy.com**. La prueba tiene por objetivo detectar la existencia de problemas en el uso del portal, en el marco de un estudio de usabilidad, a fin de mejorar la experiencia del usuario.

SE ESTÁ EVALUANDO UN PORTAL WEB, NO EL DESEMPEÑO DE USTED COMO USUARIO, POR LO TANTO, ¡NO SE PREOCUPE SI COMETE ALGÚN ERROR!

Toda la información que Ud. nos proporciona es absolutamente confidencial y muy relevante para nuestro estudio, por lo cual le agradecemos su cooperación. La prueba tiene 3 etapas:

1. En la primera etapa Ud. deberá completar un breve cuestionario con preguntas relativas a su experiencia y contexto habitual de uso de portales multimedia de video afines a **peliculashoy.com**.
2. En la segunda etapa se le proporcionará un conjunto de tareas que se deben realizar a través del portal de **peliculashoy.com**.
3. En la tercera etapa Ud. deberá completar un breve cuestionario que tiene por objetivo obtener la percepción general sobre su experiencia en el uso del portal: **peliculashoy.com**.

SI TIENE ALGUNA DUDA DURANTE EL DESARROLLO DE LA PRUEBA, USTED PUEDE PONERSE EN CONTACTO CON EL EVALUADOR.

(1) Cuestionario pre-test

Conteste el siguiente cuestionario.

I. DATOS PERSONALES

1. Sexo: Femenino Masculino

2. Edad: _____

3. Nivel de educación completado y/o en proceso:

Enseñanza media: Completa En Proceso

Técnico: Completa En Proceso

Universitario: Completa En Proceso

Otro: _____ Completa En Proceso

4. ¿Cuál es su ocupación?: _____

II. INFORMACIÓN SOBRE EXPERIENCIAS PREVIAS CON PORTALES MULTIMEDIA DE VIDEO A PELICULASHOY.COM

5. ¿Con que frecuencia accede Ud. a contenidos multimedia de video en línea, cada vez que navega por internet?

- Siempre
- Casi siempre
- Algunas veces
- Nunca

6. ¿Usted ha utilizado previamente el portal de **peliculashoy.com**?

- Si No

7. ¿Ha tenido Ud. alguna experiencia previa con los siguientes portales de películas en línea? (Puede señalar más de 1 alternativa).

- cuevana.tv
- dop.tv
- tekilaz.com
- cinefox.tv
- Youtube Movies
- Otros. ¿Cuáles? _____
- Ninguno

(2) Lista de tareas

Considere el siguiente escenario:

Usted ha tomado la decisión de emplear su tiempo libre del fin de semana, viendo una película en línea a través del portal **peliculashoy.com**. Teniendo en cuenta lo anterior, se le pide por favor realizar las tareas descritas a continuación:

Tarea 1: Buscar una película del genero Acción

1. Abra el navegador Mozilla Firefox e ingrese al portal de **peliculashoy.com**: <http://www.peliculashoy.com>
2. Encuentre la sección: "LISTA DE GENEROS" y seleccione el género "Acción".
3. Navegue por el catálogo de películas y encuentre la película "21 Black Jack".
4. Visualice la descripción de la película seleccionada, y anote la siguiente información.

Nombre de la Película	
Año de la Película	
País	
Director	

5. Debajo de la descripción de la película, escoja alguno de los posibles enlaces e inicie la reproducción. En caso de que salga alguna ventana emergente ciérrela.
6. Visualice 3 minutos de reproducción de la película escogida y vuelva a la página principal.

Tarea 2: Buscar una determinada película en el catálogo

Algunos amigos le han recomendado la película: "Casino Royale", la cual se encuentra disponible en el catálogo, Usted va a revisar la descripción de esta película para mirar si se ajusta a sus preferencias.

1. Ubique el buscador y realice una búsqueda ingresando el nombre de la película: "Casino Royale".
2. Seleccione la película encontrada e inicie la reproducción de la misma.
3. Visualice 3 minutos de reproducción de la película mencionada y responda lo siguiente:

- ¿Es de su preferencia la película sugerida?

SI NO

- Si respondió NO, indique brevemente el motivo:

Tarea 3: Buscar una película de acuerdo a un año determinado

Adicionalmente Usted echa un vistazo por las películas más recientes del portal, buscando y escogiendo una película de acuerdo a un determinado año.

1. Encuentre la sección: "Año de Estreno" y seleccione el año 2013.
2. Navegue por el catálogo de películas y encuentre la primera película cuyo País de origen no sea Estados Unidos.
3. A partir de la descripción de la película seleccionada, anote la siguiente información:

Nombre de la Película	
Año de la Película	
País	
Director	

4. Debajo de la descripción de la película, escoja alguno de los posibles enlaces e inicie la reproducción. En caso de que salga alguna ventana emergente ciérrela. Así mismo en caso de que la película no esté disponible regrese al catálogo y repita el procedimiento desde el punto 2.
5. Visualice 3 minutos de reproducción de la película escogida y vuelva a la página principal.

(3) Cuestionario pos-test

Encierre en un círculo la alternativa más apropiada para cada una de las siguientes preguntas.

1. ¿Pudo completar las tareas planteadas?

1	2	3	4	5
Muy Difícilmente	Difícilmente	Neutral	Fácilmente	Muy Fácilmente

2. ¿Considera que la información requerida en la prueba ha sido fácil de encontrar?

1	2	3	4	5
Muy Difícilmente	Difícilmente	Neutral	Fácilmente	Muy Fácilmente

3. ¿Se ha sentido orientado dentro del portal de **peliculashoy.com**?:

1	2	3	4	5
Muy en desacuerdo	En desacuerdo	Neutral	De acuerdo	Completamente de acuerdo

4. ¿Considera que es fácil navegar a través del catálogo de películas dentro del portal de **peliculashoy.com**?

1	2	3	4	5
Muy Difícilmente	Difícilmente	Neutral	Fácilmente	Muy Fácilmente

5. ¿Le parecen útiles las herramientas de búsqueda que provee el portal de **peliculashoy.com**?

1	2	3	4	5
Muy en desacuerdo	En desacuerdo	Neutral	De acuerdo	Completamente de acuerdo

6. Su grado de satisfacción en cuanto a la facilidad de encontrar una película dentro del portal es:

1	2	3	4	5
Totalmente Insatisfactorio	Poco Satisfactorio	Neutral	Satisfactorio	Muy Satisfactorio

7. Su grado de satisfacción en cuanto a la reproducción de las películas escogidas dentro del portal es:

1	2	3	4	5
---	---	---	---	---

Anexo J

J. Registros software

A continuación en este apartado se relacionan los registros software de cuatro de las aplicaciones desarrolladas en el marco del trabajo de doctorado. Estas aplicaciones software fueron registradas ante la División Nacional de Derechos de Autor (DNDA).

J1. Registros software realizados

En el presente trabajo de doctorado, se registraron cuatro de las herramientas desarrolladas, las cuales se relacionan a continuación:

- Registro del producto software titulado: “Herramienta web para la medición de estrés”, ante la Dirección Nacional de Derechos de Autor (DNDA). Libro 13, Tomo 52, Partida 136. 19 de Febrero de 2016. Esta herramienta permite la ejecución y seguimiento de pruebas de estrés sobre aplicaciones web. De este modo, a través de esta aplicación es posible obtener gráficas de las peticiones secuenciales y concurrentes realizadas sobre una determinada aplicación web.
- Registro del producto software titulado: “Herramienta para la clasificación de contenidos multimedia musicales”, ante la Dirección Nacional de Derechos de Autor (DNDA). Libro 13, Tomo 52, Partida 153. 9 de Marzo de 2016. Esta herramienta permite clasificar gráficamente un contenido multimedia musical dentro del espacio arousal-valence, para lo cual se hace uso de la librería provista por Echonest.
- Registro del producto software titulado: “Dataset de contenidos musicales de video, basado en emociones”, ante la Dirección Nacional de Derechos de Autor (DNDA). Libro 13, Tomo 54, Partida 129. 6 de Julio de 2016. El dataset registrado está en el formato JSON y reúne un conjunto de contenidos

multimedia musicales, los cuales han sido clasificados dentro del espacio arosual-valence a partir de sus características musicales.

- Registro del producto software titulado: “Sistema de recomendaciones para el servicio de VoD, basado en la inferencia de emociones a partir de variables del contexto de usuario”, ante la Dirección Nacional de Derechos de Autor (DNDA). Libro 13, Tomo 54, Partida 128. 6 de Julio de 2016. El sistema de recomendaciones registrado hace uso de variables del contexto tales como VFC Y GSR para recomendar un conjunto de contenidos musicales más adecuados a las necesidades emocionales de un usuario. Este sistema de recomendaciones fue implementado haciendo uso de la API provista por la herramienta de minería de datos weka.

A continuación se adjunta el registro obtenido de la División Nacional de Derechos de Autor para cada una de las anteriores herramientas software.



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida
13-52-136
Fecha Registro
19-feb-2016

Página 1 de 2

1. DATOS DE LAS PERSONAS

AUTOR

Nombres y Apellidos **GABRIEL CHANCHÍ GOLONDRINO** **No de identificación** 10301274
CC

Nacional de COLOMBIA

Dirección KRA 18 # 11B-56 **Ciudad:** POPAYAN

AUTOR

Nombres y Apellidos WILMAR YESID CAMPO MUÑOZ **No de identificación** 76323234
CC

Nacional de COLOMBIA

Dirección AV 19 N 15N-49. PROVITEQ UNIDAD II **Ciudad:** ARMENIA

AUTOR

Nombres y Apellidos JOSE LUIS ARCINIEGAS HERRERA **No de identificación** 76319265
CC

Nacional de COLOMBIA

Dirección TRANSV. 9N # 61BN-10, URB. BRISAS **Ciudad:** POPAYAN

PRODUCTOR

Nombres y Apellidos GABRIEL CHANCHÍ GOLONDRINO **No de identificación** 10301274
CC

Nacional de COLOMBIA

Dirección KRA 18 # 11B-56 **Ciudad:** POPAYAN

2. DATOS DE LA OBRA

Título Original HERRAMIENTA WEB DE MEDICIÓN DE ESTRÉS

Año de Creación 2015 **Pais de Origen** COLOMBIA **Año Edición**

CLASE DE OBRA INEDITA

CARACTER DE LA OBRA OBRA DERIVADA

CARACTER DE LA OBRA OBRA EN COLABORACION

ELEMENTOS APORTADOS DE SOPORTE LOGICO PROGRAMA DE COMPUTADOR

ELEMENTOS APORTADOS DE SOPORTE LOGICO MATERIAL AUXILIAR

3. DESCRIPCIÓN DE LA OBRA

LA HERRAMIENTA DE MEDICIÓN DE ESTRÉS PERMITE LA EJECUCIÓN DE PRUEBAS DE ESTRÉS SECUENCIALES Y CONCURRENTES SOBRE SITIOS WEB, GRAFICANDO EN TIEMPO REAL EL PROGRESO DE LAS PETICIONES Y GENERANDO UN LOG CON LOS RESULTADOS DE LAS PRUEBAS. LA HERRAMIENTA DE MEDICIÓN DE ESTRÉS FUE DESARROLLADA USANDO EL FRAMEWORK FLASK DE PYTHON Y LA HERRAMIENTA AB (OPERANDO EN SEGUNDO PLANO).

4. OBSERVACIONES GENERALES DE LA OBRA

5. DATOS DEL SOLICITANTE

Nombres y Apellidos **GABRIEL CHANCHÍ GOLONDRINO** **No de Identificación** **10301274**

Nacional de **COLOMBIA** **Medio Radicación** **REGISTRO EN LINEA**

Dirección **KRA 18 # 11B-56** **Ciudad** **POPAYAN**

Correo electrónico **GABRIELCHANCHI@GMAIL.COM** **Teléfono** **3008672608**

En representación de **EN NOMBRE PROPIO** **Radicación de entrada** **1-2016-11437**



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida

13-52-136

Fecha Registro

19-feb-2016

Página 2 de 2



MANUEL ANTONIO MORA CUELLAR

JEFE OFICINA DE REGISTRO

MZP

Nota: El derecho de autor protege exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. No son objeto de protección las ideas contenidas en las obras literarias y artísticas, o el contenido ideológico o técnico de las obras científicas, ni su aprovechamiento industrial o comercial (artículo 7o. de la Decisión 351 de 1993).



1. DATOS DE LAS PERSONAS

AUTOR

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de identificación	10301274
		CC	
Nacional de	COLOMBIA		
Dirección	KRA 18 # 11B-56	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	FRANCO ARTURO URBANO ORDOÑEZ	No de identificación	4616166
		CC	
Nacional de	COLOMBIA		
Dirección	--	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	DAYNER FELIPE ORDOÑEZ LÓPEZ	No de identificación	76312229
		CC	
Nacional de	COLOMBIA		
Dirección	--	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	WILMAR YESID CAMPO MUÑOZ	No de identificación	76323234
		CC	
Nacional de	COLOMBIA		
Dirección	AV 19 N 15N-49. PROVITEQ UNIDAD II	Ciudad:	ARMENIA

AUTOR

Nombres y Apellidos	ALBERTO BRAVO BUCHELI	No de identificación	97490032
		CC	
Nacional de	COLOMBIA		
Dirección	--	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de identificación	10301274
		CC	
Nacional de	COLOMBIA		
Dirección	KRA 18 # 11B-56	Ciudad:	POPAYAN

2. DATOS DE LA OBRA

Título Original HERRAMIENTA PARA LA CLASIFICACIÓN DE CONTENIDOS MULTIMEDIA MUSICALES

Año de Creación 2015 **Pais de Origen** COLOMBIA **Año Edición**

CLASE DE OBRA INEDITA

CARACTER DE LA OBRA OBRA DERIVADA

CARACTER DE LA OBRA OBRA EN COLABORACION

ELEMENTOS APORTADOS DE SOPORTE LOGICO PROGRAMA DE COMPUTADOR

ELEMENTOS APORTADOS DE SOPORTE LOGICO MATERIAL AUXILIAR

3. DESCRIPCIÓN DE LA OBRA

LA HERRAMIENTA PARA LA CLASIFICACIÓN DE CONTENIDOS MULTIMEDIA MUSICALES, PERMITE CLASIFICAR UN CONTENIDO MULTIMEDIA MUSICAL EN FORMATO MP3, DENTRO DE 8 POSIBLES EMOCIONES DE ACUERDO AL MODELO DE AROUSAL-VALENCE. LA HERRAMIENTA PARA LA CLASIFICACIÓN DE CONTENIDOS MULTIMEDIA MUSICALES FUE IMPLEMENTADA EN EL LENGUAJE DE PROGRAMACIÓN JAVA, HACIENDO USO DE LA API JEN, LA CUAL ES UNA IMPLEMENTACIÓN PARA EL LENGUAJE JAVA DEL DATASET DE CONSULTA DE CONTENIDOS MULTIMEDIA ECHONEST.

4. OBSERVACIONES GENERALES DE LA OBRA



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO

CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida

13-52-253

Fecha Registro

09-mar-2016

Página 2 de 2

5. DATOS DEL SOLICITANTE

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de Identificación	10301274
Nacional de	COLOMBIA	Medio Radicación	REGISTRO EN LINEA
Dirección	KRA 18 # 11B-56	Ciudad	POPAYAN
Correo electrónico	GABRIELCHANCHI@GMAIL.COM	Teléfono	3008672608
En representación de	EN NOMBRE PROPIO	Radicación de entrada	1-2016-16229

MANUEL ANTONIO MORA CUELLAR

JEFE OFICINA DE REGISTRO

MZP

Nota: El derecho de autor protege exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. No son objeto de protección las ideas contenidas en las obras literarias y artísticas, o el contenido ideológico o técnico de las obras científicas, ni su aprovechamiento industrial o comercial (artículo 7o. de la Decisión 351 de 1993).



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida

13-54-129

Fecha Registro

06-jul-2016

Página 1 de 2

1. DATOS DE LAS PERSONAS

AUTOR

Nombres y Apellidos	MAURICIO SÁNCHEZ BARRAGÁN	No de identificación CC	1061757023
Nacional de	COLOMBIA		
Dirección	CARRERA 3 # 6-28, BARRIO EL CENTI	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	LUIS ALEJANDRO SOLARTE MONCAYO	No de identificación CC	1061745844
Nacional de	COLOMBIA		
Dirección	BARRIO TOMAS CIPRIANO MANZANA	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de identificación CC	10301274
Nacional de	COLOMBIA		
Dirección	KRA 18 # 11B-56	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	JOSE LUIS ARCINIEGAS HERRERA	No de identificación CC	76319265
Nacional de	COLOMBIA		
Dirección	TRANSV. 9N # 61BN-10, URB. BRISAS	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	MAURICIO SÁNCHEZ BARRAGÁN	No de identificación CC	1061757023
Nacional de	COLOMBIA		
Dirección	CARRERA 3 # 6-28, BARRIO EL CENTI	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	LUIS ALEJANDRO SOLARTE MONCAYO	No de identificación CC	1061745844
Nacional de	COLOMBIA		
Dirección	BARRIO TOMAS CIPRIANO MANZANA	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de identificación CC	10301274
Nacional de	COLOMBIA		
Dirección	KRA 18 # 11B-56	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	JOSE LUIS ARCINIEGAS HERRERA	No de identificación CC	76319265
Nacional de	COLOMBIA		
Dirección	TRANSV. 9N # 61BN-10, URB. BRISAS	Ciudad:	POPAYAN

2. DATOS DE LA OBRA

Título Original DATASET DE CONTENIDOS MUSICALES DE VIDEO, BASADO EN EMOCIONES

Año de Creación	2016	País de Origen	COLOMBIA	Año Edición	
------------------------	------	-----------------------	----------	--------------------	--

CLASE DE OBRA	INEDITA
---------------	---------

CARACTER DE LA OBRA	OBRA DERIVADA
---------------------	---------------

CARACTER DE LA OBRA	OBRA EN COLABORACION
---------------------	----------------------

ELEMENTOS APORTADOS DE SOPORTE LOGICO	PROGRAMA DE COMPUTADOR
---------------------------------------	------------------------

ELEMENTOS APORTADOS DE SOPORTE LOGICO	MATERIAL AUXILIAR
---------------------------------------	-------------------



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida
13-54-129
Fecha Registro
06-jul-2016

Página 2 de 2


3. DESCRIPCIÓN DE LA OBRA

SISTEMA SOFTWARE QUE PERMITE LA GENERACIÓN Y DESCARGA DE UN LISTADO DE CONTENIDOS MUSICALES DE VÍDEO BASADO EN EMOCIONES, EL CUAL PROVEE UN ARCHIVO DESCRIPTOR JSON DE LOS CONTENIDOS Y EL TOTAL DE 200 VÍDEOS AL FINAL DE SU EJECUCIÓN.

4. OBSERVACIONES GENERALES DE LA OBRA

5. DATOS DEL SOLICITANTE

Nombres y Apellidos	MAURICIO SÁNCHEZ BARRAGÁN	No de Identificación	1061757023
Nacional de	COLOMBIA	Medio Radicación	REGISTRO EN LINEA
Dirección	CARRERA 3 # 6-28, BARRIO EL CENTRO, FRENTE A INSTITUTO PENSAR	Ciudad	POPAYAN
Correo electrónico	MAUSANBAR93@GMAIL.COM	Teléfono	3203120350
En representación de	EN NOMBRE PROPIO	Radicación de entrada	1-2016-48909


MANUEL ANTONIO MORA CUELLAR

JEFE OFICINA DE REGISTRO

MZP

Nota: El derecho de autor protege exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. No son objeto de protección las ideas contenidas en las obras literarias y artísticas, o el contenido ideológico o técnico de las obras científicas, ni su aprovechamiento industrial o comercial (artículo 7o. de la Decisión 351 de 1993).



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida

13-54-128

Fecha Registro

06-jul-2016

Página 1 de 2

1. DATOS DE LAS PERSONAS

AUTOR

Nombres y Apellidos	MAURICIO SÁNCHEZ BARRAGÁN	No de identificación	1061757023
		CC	
Nacional de	COLOMBIA		
Dirección	CARRERA 3 # 6-28, BARRIO EL CENTI	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	LUIS ALEJANDRO SOLARTE MONCAYO	No de identificación	1061745844
		CC	
Nacional de	COLOMBIA		
Dirección	BARRIO TOMAS CIPRIANO MANZANA	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de identificación	10301274
		CC	
Nacional de	COLOMBIA		
Dirección	KRA 18 # 11B-56	Ciudad:	POPAYAN

AUTOR

Nombres y Apellidos	JOSE LUIS ARCINIEGAS HERRERA	No de identificación	76319265
		CC	
Nacional de	COLOMBIA		
Dirección	TRANSV. 9N # 61BN-10, URB. BRISAS	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	MAURICIO SÁNCHEZ BARRAGÁN	No de identificación	1061757023
		CC	
Nacional de	COLOMBIA		
Dirección	CARRERA 3 # 6-28, BARRIO EL CENTI	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	LUIS ALEJANDRO SOLARTE MONCAYO	No de identificación	1061745844
		CC	
Nacional de	COLOMBIA		
Dirección	BARRIO TOMAS CIPRIANO MANZANA	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	GABRIEL CHANCHÍ GOLONDRINO	No de identificación	10301274
		CC	
Nacional de	COLOMBIA		
Dirección	KRA 18 # 11B-56	Ciudad:	POPAYAN

PRODUCTOR

Nombres y Apellidos	JOSE LUIS ARCINIEGAS HERRERA	No de identificación	76319265
		CC	
Nacional de	COLOMBIA		
Dirección	TRANSV. 9N # 61BN-10, URB. BRISAS	Ciudad:	POPAYAN

2. DATOS DE LA OBRA

Título Original	SISTEMA DE RECOMENDACIONES PARA EL SERVICIO DE VOD, BASADO EN LA INFERENCIA DE EMOCIONES A PARTIR DE VARIABLES DEL CONTEXTO DE USUARIO		
------------------------	--	--	--

Año de Creación	2016	País de Origen	COLOMBIA	Año Edición	
------------------------	------	-----------------------	----------	--------------------	--

CLASE DE OBRA	INEDITA
----------------------	---------

CARACTER DE LA OBRA	OBRA DERIVADA
----------------------------	---------------

CARACTER DE LA OBRA	OBRA EN COLABORACION
----------------------------	----------------------

ELEMENTOS APORTADOS DE SOPORTE LOGICO	PROGRAMA DE COMPUTADOR
--	------------------------

ELEMENTOS APORTADOS DE SOPORTE LOGICO	MATERIAL AUXILIAR
--	-------------------



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO

CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida

13-54-128

Fecha Registro

06-jul-2016

Página 2 de 2

3. DESCRIPCIÓN DE LA OBRA

SISTEMA WEB RECOMENDADOR DE CONTENIDOS MULTIMEDIA DE VIDEO MUSICALES, EL CUAL CONSIDERA VARIABLES DEL CONTEXTO DE USUARIO COMO VARIABLES FISIOLÓGICAS (VFC, GSR Y EMG), MEDIDAS QUE SIRVEN PARA DETERMINAR EL ESTADO DE ANIMO QUE VA AYUDAR EN LA CLASIFICACIÓN Y RECOMENDACIÓN DE LOS CONTENIDOS.

4. OBSERVACIONES GENERALES DE LA OBRA

5. DATOS DEL SOLICITANTE

Nombres y Apellidos	MAURICIO SÁNCHEZ BARRAGÁN	No de Identificación	1061757023
Nacional de	COLOMBIA	Medio Radicación	REGISTRO EN LINEA
Dirección	CARRERA 3 # 6-28, BARRIO EL CENTRO, FRENTE A INSTITUTO PENSAR	Ciudad	POPAYAN
Correo electrónico	MAUSANBAR93@GMAIL.COM	Teléfono	3203120350
En representación de	EN NOMBRE PROPIO	Radicación de entrada	1-2016-48635



MANUEL ANTONIO MORA CUELLAR

JEFE OFICINA DE REGISTRO

MZP

Nota: El derecho de autor protege exclusivamente la forma mediante la cual las ideas del autor son descritas, explicadas, ilustradas o incorporadas a las obras. No son objeto de protección las ideas contenidas en las obras literarias y artísticas, o el contenido ideológico o técnico de las obras científicas, ni su aprovechamiento industrial o comercial (artículo 7o. de la Decisión 351 de 1993).