

CARATULA



**UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA**  
*La Universidad Católica de Loja*

**ÁREA TÉCNICA**

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS Y  
COMPUTACIÓN

**Ontología y aplicación web de mantenimiento para emociones humanas en  
videos de YouTube**

TRABAJO DE FIN DE TITULACIÓN.

AUTOR: Plascencia Arévalo, Yoder Santiago

DIRECTOR: Gómez Alvarado, Héctor Fernando, PhD

LOJA – ECUADOR

2014

## **APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN**

PhD. Héctor Fernando Gómez Alvarado.

**DOCENTE DE LA TITULACIÓN**

De mi Consideración:

El presente trabajo de fin de titulación: Ontología y aplicación web de mantenimiento para emociones humanas en videos de YouTube realizado por Plascencia Arevalo Yoder Santiago; ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, abril de 2014

f) .....

## DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo, Plascencia Arevalo Yoder Santiago, declaro ser autor del presente trabajo de fin de titulación: Ontología y aplicación web de mantenimiento para emociones humanas en videos de YouTube, de la titulación de Ingeniería en sistemas y ciencias de la educación, siendo el PhD. Héctor Gómez director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

f. ....

Autor: Plascencia Arevalo Yoder Santiago

Cédula: 1104635519

## **DEDICATORIA**

Dedico el presente trabajo a mis padres por todo el apoyo y la guía que brindado en cada una de las etapas que me han llevado a culminar este escalón de mi vida académica; por todo el esfuerzo y buen ejemplo impartido en las diferentes dimensiones que componen mi persona; a todos los docentes con los que tuve la suerte de compartir mi vida universitaria en especial a aquellos que supieron desafiar mis capacidades para incentivar en mi los deseos de superarme y destacarme en mi carrera.

## **AGRADECIMIENTO**

Agradezco en primer lugar a mis padres y mi hermano por todo el amor, cariño y comprensión que me brindaron durante toda mi vida académica, por darme fuerzas cuando sentía desfallecer en el camino y por darme esperanza tras cada tropiezo; a mi director de tesis por confiar en mi capacidad y apoyarme durante el desarrollo del presente trabajo, el cual no habría sido posible sin su colaboración; a todos mis compañeros programadores del Club Java con los que compartimos grandes experiencias que han dado luz a mi vocación profesional; a todos los docentes con los que compartí ciclo tras ciclo los diferentes componentes académicos de mi carrera, por el conocimiento recibido de ellos para mi desarrollo integral como profesional; finalmente agradezco a mis compañeros por todos los recuerdos, malas noches compartidas, sesiones de estudio y demás actividades del día a día universitaria.

## ÍNDICE DE CONTENIDOS

CARATULA.....	i
APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN .....	ii
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS .....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
ÍNDICE DE CONTENIDOS.....	vi
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
DOMINIO ONTOLÓGICO.....	5
1. El problema .....	6
2. La fuente de Datos .....	7
3. Reutilización Ontológica .....	8
3.1. Ontología VERL .....	8
3.1.1. Dominio Ontológico.....	8
3.1.2. Descripción Ontológica .....	9
3.2. Ontología OWL-time.....	10
3.2.1. Dominio Ontológico.....	10
3.2.2. Descripción Ontológica .....	11
3.3. Otros Recursos Ontológicos Relacionados .....	11
4. Metodología.....	11
4.1. NeOn Methodology .....	12
4.2. Metodología de Uschold y King.....	14
4.3. Metodología modular de Hois, Bhatt, & Kutz .....	17
5. Evaluación Metodológica.....	20
MODELADO ONTOLÓGICO.....	24
1. Modelo conceptual .....	25
2. Capa Cualitativa .....	27
3. Capa Cuantitativa .....	30
4. Capa Axiomática .....	31
5. Resultado del Modelado Ontológico.....	34

APLICACIÓN DE MANTENIMIENTO ONTOLÓGICO .....	39
1. Proceso General y Arquitectura .....	40
1.1. Software Ontológico en Monitoreo y Control.....	40
1.2. Diseño del sistema .....	43
2. Herramientas .....	45
2.1. Jena.....	45
2.2. Api de YouTube.....	45
2.3. Extracción de subtítulos .....	45
3. Desarrollo .....	47
3.1. Marco de trabajo de desarrollo y redes ontológicas.....	47
3.2. Sentencias SPARQL .....	48
4. Resultado del Aplicativo .....	51
4.1. Mantenimiento de Instancias de Emociones en Videos.....	51
4.2. Exportación de Datos .....	52
5. Trabajos Futuros .....	53
CONCLUSIONES.....	54
RECOMENDACIONES.....	56
BIBLIOGRAFIA.....	57
ANEXOS.....	61
1. Anexo A .....	62
2. Anexo B.....	71
3. Anexo C.....	77
4. Anexo D.....	81
5. Anexo E .....	87

## RESUMEN

El presente trabajo muestra el diseño y elaboración de una ontología para el mantenimiento y gestión de datos de interés en videos de YouTube, principalmente emociones de estrés humano; se emplea recursos ontológicos en todo el proceso de desarrollo para consolidar un componente de software que ejemplifica el uso de la ontología desarrollada, teniendo como principal función crear una base ontológica para el entrenamiento de agentes inteligentes en sistemas de vigilancia; la herramienta brinda la funcionalidad de exportar los datos de las emociones registradas en formatos como: RDF/XML, RDF/XML-ABBREV, RDF/JSON, N3, N-TRIPLE, EXCEL.

Se hace uso del lenguaje java y su marco de trabajo web JSF para integrar las librerías de Jena que se encargan del manejo RDFS mediante SPARQL y una unidad html para la implementación de un navegador virtual en el servidor que permita extraer subtítulos de los videos; se emplean técnicas de Ajax, recursos de html5 y programación del lado del cliente en JavaScript para realizar peticiones asíncronas que permite mejorar la experiencia del usuario.

**PALABRAS CLAVE:** Videos, Emociones, Estrés, Ontología, RDF, Jena, Java, Sparql, Youtube.



## **ABSTRACT**

This document presents the design and development of an ontology for maintenance and management of interest data in YouTube videos , mainly human stress emotions; ontological resources are used throughout the development process to consolidate a software component capable for demonstration purposes; the main function is to create an ontological base for training intelligent agents in surveillance systems, the tool provides the functionality to export registered emotions data in different formats such as: RDF/XML, RDF/XML-ABBREv, RDF/JSON , N3 , N-TRIPLE , EXCEL .

Based of java language and his JSF web framework; the development process includes libraries like Jena, who runs the operation with RDFS using the SPARQL specifications and an html component for implementing a virtual browser on the server side that allows extracting subtitles of the videos; the use of techniques such as Ajax, html5 resources and client-side programming in JavaScript, for asynchronous requests improves the user experience.

**KEYWORDS:** Videos, Emotions, Stress, Ontology, RDF, Jena, Java, SPARQL, YouTube.

## INTRODUCCIÓN

El trabajo realizado estuvo enfocado en videos y emociones humanas para dar respuesta a la demanda de sistemas de seguridad inteligentes; para desarrollar sistemas de esta naturaleza se requiere de inteligencia artificial y mecanismos para entrenar agentes y redes neuronales en tareas de reconocimiento e inferencia, como determinar si un rostro puede ser un indicador de una potencial amenaza; en resumen, se buscó una manera inteligente de acumular conocimiento para posterior entrenamiento.

En el primer capítulo de este trabajo se detalla el dominio del problema y se describe los trabajos y recursos ontológicos empleados en el proceso de diseño de la ontología para el registro de emociones. En el segundo capítulo, se describe la metodología utilizada en el proceso de creación de la ontología de emociones humanas en videos de YouTube. El tercer capítulo describe una aplicación realizada para dar mantenimiento a las instancias de la ontología utilizando la librería Jena para la construcción de consultas SPARQL utilizadas en el desarrollo de un módulo que permita manipular el fichero RDF/XML; también explica la arquitectura que se utilizó para el desarrollo del aplicativo mediante el lenguaje de programación java, el proceso empleado para extraer los subtítulos de los videos de YouTube; también de explica el desarrollo de un componente de exportación para los datos almacenados en la ontología y los resultados de todo el desarrollo integrados en una aplicación web.

La necesidad de realizar una aplicación que use un archivo RDF en lugar de una base de datos, responde a un problema muy particular; una base de datos no es la manera más adecuada de representar el orden real de las cosas; es decir, aunque la base de datos sea rápida, soporte concurrencia y sea altamente transaccional, se requieren técnicas muy complejas de minería de datos para extraer nuevo conocimiento a partir de sus registros; no así una ontología, donde los datos se almacenan en estructuras de nodos y se puede inferir nuevo conocimiento basado en axiomas.

Como objetivo general se planteó diseñar una manera inteligente de crear recursos de entrenamiento en reconocimiento de emociones humanas; como objetivos específicos se propuso: Demostrar la capacidad de una lógica descriptiva para gestionar un modelo de datos de manera consistente e inteligente, al punto de funcionar como un acercamiento a una base de conocimiento. Utilizar de manera exitosa una arquitectura para una aplicación web basada en ontologías. Encontrar una manera eficiente de obtener en formato texto los enunciados y expresiones de las personas en los videos de YouTube para relacionarlas con las emociones que se parecían. Hacer uso de tecnologías recientes de desarrollo web para

conseguir una experiencia de usuario agradable y un frontal que responda a los principales atributos de calidad. Se logró dar cumplimiento a todos los objetivos antes mencionados.

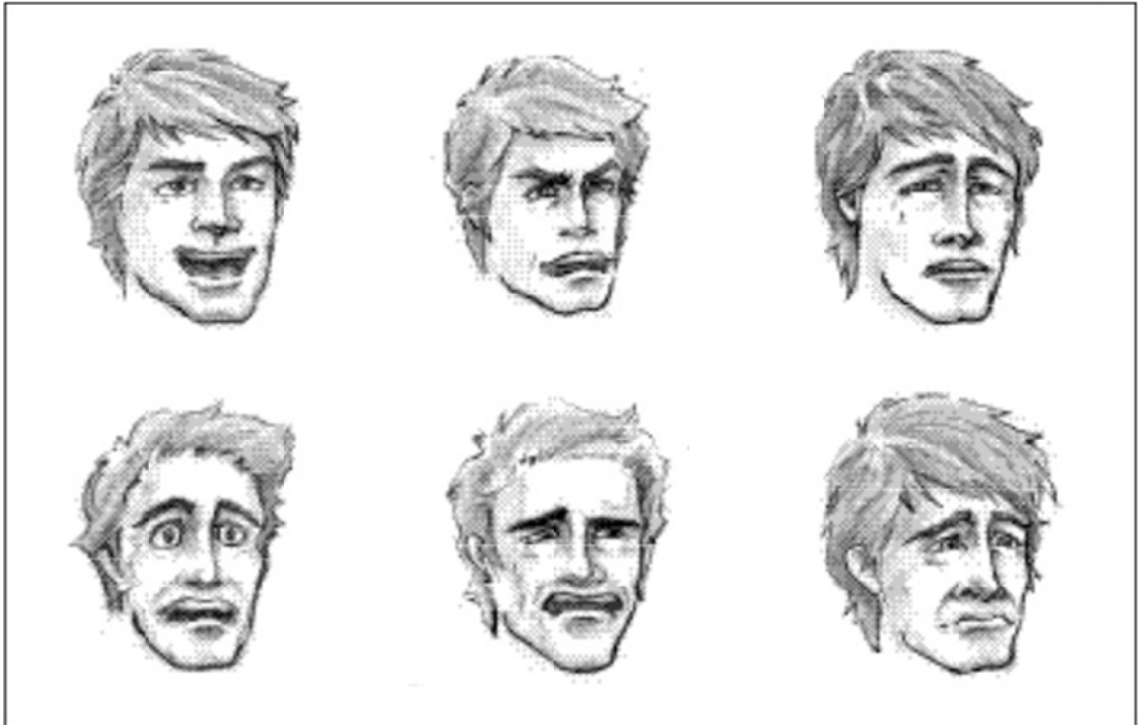
Se desarrolló una ontología que permita consumir información de YouTube; particularmente se hace énfasis en rostros humanos y las emociones que se pueden apreciar en ellos; además se extrae los subtítulos si los hay, para relacionar emociones con expresiones verbales. El desarrollo incluye un aplicativo web que le permite a un operador ingresar un video y registrar emociones que puedan apreciarse en determinado instante del video; toda la información, es almacenada en un archivo en formato RDF/XML que funciona a manera de base de datos; se desarrolló además un módulo de reportes que permite, exportar los datos por emoción en diferentes formatos para su análisis y explotación en propósitos de entrenamiento o como entradas para futuros componentes de software relacionado al reconocimiento de emociones en recursos de video.

Se encontraron diversas dificultades como el problema de la flexibilidad del modelado ontológico; si bien es cierto resulta enormemente útil que se pueda modelar cualquier lógica, es necesario recurrir a estándares de modelado y aun haciendo uso de ellos, la ontología puede no ser entendida por todas las herramientas de modelado ontológico disponibles en el mercado. Otro inconveniente en el desarrollo fue el tema del lenguaje, una emoción en un rostro podría considerarse universal; no así el idioma, ya que una cadena de texto que representa una palabra, tiene contexto geográfico y social; YouTube posee subtítulos en diferentes idiomas, se decidió trabajar sobre el inglés por ser el más abundante en videos de YouTube. El problema más complejo resultó ser en el ámbito técnico del aplicativo del mantenimiento de instancias de emociones humanas, inicialmente se pretendía separar los canales de audio y video del recurso multimedia para utilizar el audio y procesarlo con un reconocedor de voz instalado en el servidor, pero esta solución generó demasiada espera en el cliente; luego se intentó hacer uso de los complementos propios del navegador de Google, pero esta idea se descartó por que la aplicación dependería de un solo navegador para poder funcionar; desde el principio se descartó el uso del API de YouTube, únicamente se puede acceder a subtítulos de videos del titular de una cuenta de Google; esto limita enormemente las fuentes de datos a solo los videos del canal propio del operador del sensor; la solución más eficiente fue realizar técnicas de virtualización para crear un navegador en el servidor que pudiera extraer los subtítulos desde el código HTML de la página del video.

## **DOMINIO ONTOLÓGICO**

## 1. El problema

Al observar un rostro humano, puede resultar muy complicado emitir un criterio respecto a lo que la persona está sintiendo; la misma expresión facial puede significar algo diferente para determinados observadores, un rostro humano no puede denotar una emoción con tanta claridad como para brindar al observador, seguridad al emitir un criterio; si se toma en consideración lo que la persona está diciendo en el momento de apreciar la emoción, un observador podrá identificar con mayor efectividad determinada emoción.



**Ilustración 1.** Felicidad, ira, tristeza, sorpresa, disgusto y miedo  
**Fuente:** (Parker, 2010)

Si para un ser humano que esta supuesto a entender las emociones humanas de manera instintiva, a veces resulta muy complicado identificarlas, es más complicado aún para los sistemas actuales en diversos ámbitos de la sociedad; al considerar el funcionamiento de los sistemas de seguridad actuales por ejemplo, existen muchos puntos de falla asociados a la necesidad de un operador humano para lanzar o no una alerta en función de lo apreciado a través de una cámara de vigilancia, si se logra eliminar el factor humano o al menos se brinda herramientas automáticas para el reconocimiento de situaciones peligrosas, se evitarían toda clase de incidentes de seguridad. En la era de web social la cantidad de información que los usuarios generan puede ser utilizada para analizar tendencias sociales y personales que en otros tiempos implicaban complicados métodos de muestreo poblacional y los resultados podían resultar muy tardíos e ineficientes, con la capacidad de identificar

emociones humanas en fotos y videos en perfiles sociales, de manera automática podríamos empezar a identificar patrones de conducta complejos para determinar cuando la depresión puede tornarse en suicidio o la ira en asesinato. La identificación de emociones tiene amplias aplicaciones en términos de comercio, si una cámara detectara que un cliente sonríe al ver un producto y no lo compra, se podría recomendar el producto a la persona en la caja, el hecho de saber que determinado producto produce alegría en un consumidor, aumenta la probabilidad de vérselo. De manera general se puede decir que cualquier ámbito de la sociedad puede ser mejorado con el reconocimiento emocional, si la nueva generación de interfaces humanas y sistemas inteligentes tiene la capacidad de diferenciar las emociones de sus usuarios, se puede especializar la manera en que los sistemas interactuar con el usuario para mejorar su experiencia.

Resulta muy útil saber de qué se está hablando en un video al momento de determinar la emoción relacionada; por tal motivo, se introdujo el concepto subtítulo; el reconocimiento de emociones basado en el estudio de proporciones faciales como lo propone M.H. Bindu (2007), debe ser complementado con el reconocimiento automático en base a entrenamiento y contexto; toda información relacionada a la persona en el instante en que se aprecia la emoción, es relevante para la decisión del estado emocional apreciado; un rostro humano gritando no significa lo mismo si al gritar usa la palabra gol que usar la palabra auxilio; fue necesario crear un acercamiento a una base de conocimiento particularmente sobre emociones de estrés humano; es decir, emociones que se aprecian en circunstancias que exigen una respuesta extrema del ser humano, por tal motivo se introdujo el concepto de emoción de estrés.

## **2. La fuente de Datos**

Es necesario llevar la tecnología al reconocimiento de emociones, por la complejidad del problema la solución inequívoca implica la utilización de recursos de inteligencia artificial; no es posible preparar datos de prueba para todos los escenarios en los que se puede dar el reconocimiento de una emoción, la respuesta se encuentra en los millones de videos que se encuentran disponibles en Internet, específicamente en YouTube; a partir del conocimiento que se pueda extraer de los videos en línea se puede generar recursos de entrenamiento para generar sistemas capaces de identificar emociones en toda clase de circunstancias.

Un video de YouTube no es la única fuente de la cual se pueda apreciar emociones humanas, podemos considerar además que existen imágenes públicas en redes sociales, blogs y sitios web; el medio de adquisición de datos utilizado es YouTube, debido a que en la gran mayoría de sus videos se encuentra la posibilidad de extraer los subtítulos.

### **3. Reutilización Ontológica**

Para la creación de una ontología se han venido utilizando lenguajes de lógica descriptiva como RDF/XML y OWL, también conocidos como lenguajes de predicados, permiten estructurar el conocimiento de una forma más inteligente para las máquinas; una ontología sería el equivalente a un modelo de datos en el desarrollo tradicional de aplicaciones y requiere un tratamiento diferente.

Los sistemas de información generan una gran cantidad de datos, estos datos se utilizan para diversos propósitos y se visualizan en toda clase de dispositivos; la forma en que los datos se estructuran responde en gran medida a la información que se quiere obtener de ellos; pero cuando se quiere obtener conocimiento se emplea estructuras de datos enlazados, los datos enlazados comprenden la creación de vínculos mediante la web; uno de los principios básicos de los datos enlazados apunta a la reutilización de definiciones de diversos orígenes (Berners-Lee, Bizer, & Heath, 2009); para respetar estos, se debe tratar de reutilizar diccionarios ontológicos que permitan modelar una aplicación para gestionar datos de videos de YouTube; en los apartados siguientes se describen algunas metodologías para realizar el diseño final de la ontología de emociones en videos de YouTube y un aplicativo de gestión para los datos de interés.

#### **3.1. Ontología VERL**

La ontología VERL<sup>1</sup> describe una serie de conceptos, entidades y relaciones necesarias para el modelamiento de objetos físicos y sus actividades para el reconocimiento automático en videos (Bremond, Maillot, Thonnat, & Vu, 2004); VERL propone también la sintaxis de un lenguaje de representación ontológica para describir las estructuras propuestas.

##### **3.1.1. Dominio Ontológico**

La consolidación de una ontología para el reconocimiento de actividades en recursos multimedia como videos, resulta muy complicado; desarrolladores y expertos relacionados al dominio, tienen diferentes percepciones de como es el comportamiento de un objeto físico, más aun tratándose de seres humanos; las distintas percepciones de las realidades físicas llevan a generar múltiples representaciones ontológicas de los objetos; existen otras variables que dificultan la tarea de reconocimiento y que no están relacionadas con la descripción de los objetos; en acciones y expresiones humanas por ejemplo, cada región del mundo posee conductas propias de la realidad del país al que pertenecen. La terminología seleccionada para la descripción de las relaciones y conceptos en el contexto de análisis de video, son tomados de la vida diaria y esto genera toda clase de problemas en términos de

---

<sup>1</sup> Video Events Representation Language

ambigüedad, por tal motivo el enfoque propuesto por Francois Bremond et al. (2004) apunta a identificar realidades físicas muy puntuales dentro de una secuencia de video normal.

Para el contexto de apreciación de emociones, se requiere identificar una emoción en el rostro de una persona; los métodos para identificar emociones radican en las proporciones que existen en los elementos que componen el rostro humano; M.H. Bindu (2007) en su artículo demuestra un sistema que usa algoritmos de proporción facial para el reconocimiento basado en técnicas de inteligencia artificial, pero el proceso de reconocimiento puede mejorarse enormemente si además del uso de algoritmos de proporciones faciales, se usan datos de entrenamiento bastante especializados y jerarquías para identificar emociones complejas; se creó una ontología para generar una fuente de datos que a futuro servirá para el entrenamiento de agentes inteligentes en tareas de reconocimiento de emociones humanas.

### 3.1.2. Descripción Ontológica

Para la creación del recurso ontológico, se reutilizaron tres definiciones de la ontología propuesta por Francois Bremond et al. (2004); el primer concepto seleccionado es el objeto físico que se refiere a toda entidad material apreciable dentro de un video pudiendo ser una persona, un animal, un juguete, un árbol o cualquier objeto. La clase del objeto físico observado tiene que ser especificada de tal manera que las acciones observadas sean propias de cada objeto.

El segundo concepto reutilizado es el estado primitivo que se refiere a cualquier suceso observable en un determinado instante; se debe definir los estados primitivos como clases independientes del objeto observado, la misma acción puede ser reutilizada por más de un objeto. Al relacionar un objeto físico y un estado primitivo surge un evento primitivo que representa el tercer concepto reutilizado de la ontología propuesta por Francois Bremond et al. (2004); en el Anexo B se puede encontrar un informe descriptivo con los elementos reutilizados.

```
PrimitiveEvent Changes_zone
physical_objects:
( (p : Person), (z1 : Zone), (z2 : Zone) )
componants:
( (c1 : PrimitiveState Inside_zone(p, z1))
(c2 : PrimitiveState Inside_zone(p, z2)) )
constraints:
//Sequence
(c1; c2)
```

**Ilustración 2.** Ejemplo de evento simple VERL.

**Fuente:** (Bremond, Maillot, Thonnat, & Vu, 2004).



Según la ontología propuesta por Francois Bremond et al. (2004) un cambio de zona es un evento simple con dos componentes identificables, la persona es identificada en una zona 1 y luego en una zona 2, la única restricción para este evento es que los componentes deben ser secuenciales y nunca pueden suceder en el mismo instante.

En el instante en que una emoción es capturada en un video de YouTube la persona puede estar hablando de algún tópico en particular; si se relaciona la emoción expresada con el contexto de su conversación se puede mejorar el reconocimiento de las emociones; en resumen se trata de un evento simple apreciado en un video, que es manifestar emociones mientras se habla de un tema.

Cuando se logra identificar un rostro humano, se puede apreciar la emoción que transmite; la interpretación de una emoción puede ser determinada por el lugar en el que se está realizando el evento, la ontología propuesta por Francois Bremond et al. (2004) permite modelar este aspecto como un evento primitivo.

```
PrimitiveEvent shows
physicalObjects
( (p:Person) )
components
p:Person show_a emotion
```

**Ilustración 3.** Captura de emociones en notación (Bremond, Maillot, Thonnat, & Vu, 2004).  
**Fuente:** Propia.

## 3.2. Ontología OWL-time

### 3.2.1. Dominio Ontológico

El reconocimiento de un evento en un video puede estar sujeto a muchas condiciones temporales, se puede considerar que un evento capturado en video sucedió en determinada fecha, la apreciación del fenómeno sucede en otra fecha y hora diferentes, el evento observado en el video, tiene un contexto temporal en función de la duración del video; las diferentes dimensiones temporales atadas al reconocimiento de emociones humanas en videos de YouTube, hacen necesario introducir entidades temporales complejas; permitiendo a la ontología describir el fenómeno de una manera más apegada a la realidad; las dimensiones temporales atadas a un suceso, pueden ser modeladas mediante la ontología OWL-time (W3C, 2006).

La dimensión temporal inherente a la apreciación de una emoción en un video, corresponde a un instante específico y no un evento compuesto por una sucesión de instantes; describir el instante en el que se aprecia una emoción en un video, implica expresar el evento en

términos de hora, minutos y segundos; para referirse al cuadro del video en el que sucede la emoción, se emplea la definición instante de la ontología OWL-Time.

### 3.2.2. Descripción Ontológica

A partir de la ontología propuesta por Francois Bremond et al. (2004) se introdujo el concepto de evento primitivo para describir fenómenos apreciados en videos, la apreciación de una emoción sucede en un determinado instante del video, por lo tanto un evento primitivo está sujeto a condiciones temporales; se puede decir que un evento primitivo sucede en un instante determinado, para la ontología temporal OWL-Time cualquier concepto que determine una dimensión temporal, proviene de una clase llamada entidad temporal.

```
:TemporalEntity  
  a owl:Class ;  
  rdfs:subClassOf .
```

**Ilustración 4.** Entidad temporal de Owl Time.  
**Fuente:** (W3C, 2006).

Un evento primitivo sucede en un determinado instante; se puede relacionar los conceptos evento primitivo de la ontología VERL, con el instante definido en la ontología OWL-Time, de este modo las complejas definiciones temporales que posteriormente puedan relacionarse a la emoción, se pueden expresar bajo las definiciones y restricciones ya introducidas por la ontología. En el Anexo C se puede encontrar un resumen descriptivo acerca de la ontología OWL-Time para definición de tiempo.

```
:Instant  
  a owl:Class ;  
  rdfs:subClassOf  
  :TemporalEntity .
```

**Ilustración 5.** Instante de Owl Time.  
**Fuente:** (W3C, 2006).

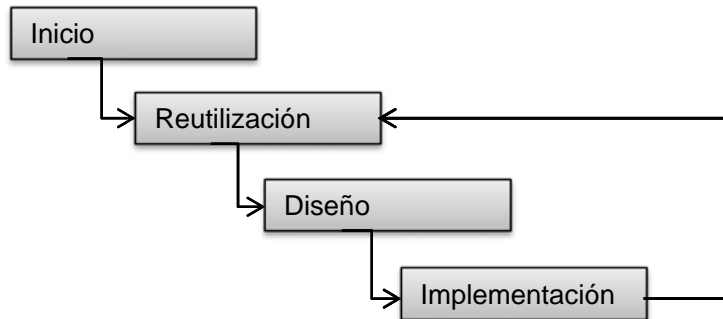
### 3.3. Otros Recursos Ontológicos Relacionados

## 4. Metodología

Para la creación de la ontología de emociones humanas en videos de YouTube se seleccionó la metodología propuesta por Joana Hois et al. (2009); en esta sección se encuentra un resumen de las metodologías candidatas y una evaluación de cada una de ellas en función del dominio ontológico del presente trabajo; se describe también los criterios técnicos utilizados para sustentar la utilización de la metodología seleccionada.

#### 4.1. NeOn Methodology

Existe una gran similitud entre las metodologías tradicionales de desarrollo y el marco de trabajo propuesto en NeOn Methodology, siendo lo suficientemente flexible como para funcionar como un modelo en cascada de entre 4 y 6 etapas. Según las características del ámbito de negocio se estructuran las etapas necesarias para satisfacer las necesidades del cliente, donde la versión más simple del proceso de desarrollo, consiste de 4 etapas que son: inicio, reutilización, diseño e implementación.



**Ilustración 6.** Modelo Cascada de 4 etapas Methontology.  
Fuente: (NeOn Proyect, 2006).

Además del modelo en cascada (NeOn Proyect, 2006) también propone un ciclo de vida iterativo e incremental para el desarrollo ontológico, básicamente agrupa pequeños ciclos de cascada en  $n$  iteraciones según el proyecto lo requiera; haciendo muy libre la forma en la que la metodología es aplicada y dotándola de la capacidad de adaptarse a equipos que trabajen bajo paradigmas iterativos o en cascada.

Al igual que las metodologías tradicionales; NeOn Methodology en cada una de sus fases genera artefactos muy similares a los generados en ciclo de RUP. Si se considera el hecho de que se manejan los mismos conceptos de metodologías tradicionales,; existen 9 escenarios de utilización para NeOn Methodology y que deben ser considerados para la aplicaciones (Gómez-Pérez & Suárez-Figueroa, 2009).

**Tabla 1.** Escenarios de uso Methontology.

Escenario	Descripción
De especificación a Implementación	Comienza con actividades de adquisición de conocimiento de la organización que resulta en especificaciones de requerimientos; no existe reutilización de recursos.
Reutilización y reingeniería de recursos no ontológicos	Las especificaciones de requerimientos son construidas a partir de recursos no ontológicos que puedan ser reutilizados y transformados para la creación de ontologías.
Reutilización de recursos ontológicos	Se reutiliza recursos ontológicos ya construidos para crear redes ontológicas o módulos para sistemas ontológicos.

Reutilización y reingeniería de recursos ontológicos	Según los requerimientos se puede reusar y modificar recursos ontológicos para la creación de módulos o extensión de redes ontológicas.
Re-uso y combinación de recursos ontológicos	Según los requerimientos se puede reusar recursos ontológicos y crear nuevos relacionando definiciones existentes para la creación de módulos o extensión de redes ontológicas.
Reutilización, combinación y reingeniería de recursos ontológicos	Se requiere reutilizar varios recursos ontológicos de los cuales se combinan conceptos para ajustarse a los requerimientos; además implica el rediseñar recursos ya existentes.
Reutilización de patrones de diseño ontológico	Existen directorios públicos que agrupan patrones de diseño ontológicos que pueden ser reutilizados para la creación de recursos ontológicos nuevos.
Reestructuración de recursos ontológicos	En este escenario ya existen recursos ontológicos creados y el desarrollador extiende definiciones, especializa nodos o reestructura módulos de el aplicativo.
Localización de recursos ontológicos	En este escenario se debe considerar el contexto del recurso ontológico, idioma, cultura, país, etc. Se orienta un recurso a determinado contexto y se limita su alcance.

**Fuente:** (Gómez-Pérez & Suárez-Figueroa, 2009).

El segundo escenario de NeOn Methodology refleja mejor la problemática de la creación de una ontología para videos de YouTube, se requiere crear una ontología a partir de recursos no ontológicos como los son los subtítulos y las emociones que se pueden observar en los videos; también es pertinente buscar y reutilizar recursos ontológicos que representen elementos como video, tiempo y subtítulos.

NeOn Book<sup>2</sup> publicado en el sitio web de NeOn Project (2006) presenta un marco de trabajo muy completo para la creación de redes ontológicas; su similitud con metodologías tradicionales de desarrollo garantizan un fácil entendimiento del proceso de levantamiento y licitación de requerimientos, pero se puede incurrir en abstracciones demasiado cercanas al modelamiento tradicional, provocando que el modelo resultante esté más orientado a la satisfacción del cliente/patrocinador que a reflejar el orden real de las cosas.

Otro inconveniente de emplear NeOn Methodology es el proceso de desarrollo como tal; el marco de trabajo define el modelado y deja abierto el proceso de codificación, quedando a criterio del desarrollador como llevar el modelado al código; el simple hecho de pasar de un espacio conceptual al otro; es decir de ontología a código, genera toda clase de incongruencias que ingenieros Ontológicos y programadores no son capaces de anticipar;

---

<sup>2</sup> Documento que describe los procedimientos de Methontology.

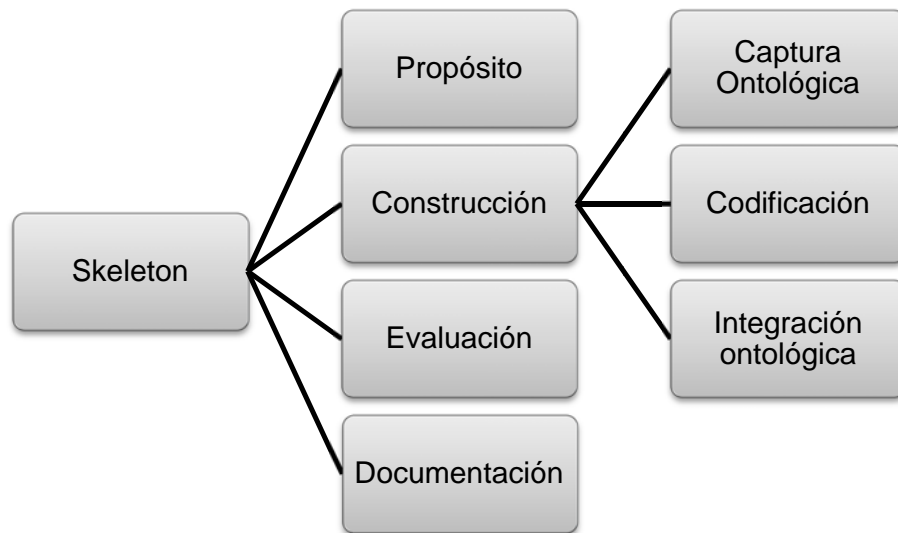
un ejemplo clásico de este tipo de problemas es la herencia de atributos, en los lenguajes orientados a objetos, no existe tal concepto ya que solo se puede heredar por medio de clases o implementar comportamientos por medio de interfaces; otro problema típico es el manejo de restricciones, cuando un programador debe codificar el nivel axiomático de una ontología, las reglas que garantizan la consistencia del modelado ontológico son parte de un mismo dominio, pero en términos de código una ontología no es suficiente para determinar si una restricción se coloca a nivel de entidad, aplicación, pantalla, capa de reglas de negocio, etc. Debido a la alta cohesión que se requiere en un aplicativo, el modelado ontológico puede verse perjudicado; se debe considerar además que a partir de una ontología, el programador no puede decidir la estructura lógica del aplicativo, la colocación de utilitarios, la paginación de consultas, la cantidad de información por canal, utilización de mapeo relacional u objetos de acceso a datos, tipos de servicios expuestos, encapsulación de atributos y demás detalles que no son relevantes en una ontología pero que pueden llevar a un aplicativo al total fracaso.

#### **4.2. Metodología de Uschold y King**

En 1995 ya existía un cuerpo de conocimiento bastante extenso sobre ontologías; la gran mayoría de conocimiento generado respondía a inquietudes muy específicas sobre las ontologías como: definición, propósito, aplicación, lenguaje, estructura, herramientas de creación, etc. Eran muy escasos los recursos académicos respecto a cómo crear una ontología, es decir una metodología de creación de ontologías (Uschold & King, 1995).

El proceso de creación de una ontología puede arrojar mucho conocimiento sobre su naturaleza; el criterio y experiencia del ingeniero ontológico determina el conocimiento introducido en las definiciones que componen un recurso ontológico; sin embargo existen elementos que no son susceptibles al criterio del diseñador, procedimientos genéricos comunes a todo diseño ontológico; surge la metodología Uschold & King (1995) que propone 4 etapas para la creación de recursos ontológicos como un esqueleto, sobre las 4 etapas básicas se puede extender el marco de trabajo según lo demande el dominio de negocio o el juicio experto.

Si hacemos un símil con las metodologías de desarrollo tradicionales, podemos encontrar que según los postulados propuestos por Philippe Kruchten (2004), la metodología de Mike Uschold y Martin King (1995), es muy parecida a un modelo cascada bastante simplificado, pero es importante tener en cuenta que se trata de un esqueleto para el proceso de creación de ontología y que la metodología puede extenderse si la naturaleza del negocio así lo exige.



**Ilustración 7.** Fases metodológicas de Uschold y King.

**Fuente:** (Uschold & King, 1995).

El propósito apunta a tener bien identificado el uso y el rango de intereses de los usuarios de la ontología; existen descripciones muy completas sobre el ámbito de aplicación y propósito de varias ontologías; según Mike Uschold et al. (1995) vale la pena considerar estos recursos, aunque no se puedan reutilizar completamente pueden agilizar el proceso de diseño; al reutilizar definiciones ontológicas se está cumpliendo con los principios básicos de los datos enlazados. El propósito también se ve afectado por la naturaleza del software para el cual se está diseñando; el desarrollo de un recurso ontológico puede apuntar a la creación de una base de conocimiento donde el proceso de diseño y validación de los datos en la ontología puede implicar muchos recursos y atención particular en la consistencia de los datos; como acercamiento de una base de conocimiento se apunta al consumo de la información y es vital que la información sea correcta, un ejemplo de ello es DBpedia<sup>3</sup> con toda clase de información orientada a ser usada como un buscador. De manera opuesta a una base de conocimiento cuando se quiere crear una ontología para un caso mucho más particular, el ámbito de modelamiento apunta más a la extracción de conocimiento a partir de la interacción con usuarios, un ejemplo de este escenario lo proponen Fonou-Dombeu & Huisman (2011) para un sistema de asistencia técnica para ayudar a mujeres con problemas de acceso a tecnologías de la información; el dominio del problema no requiere trabajo de modelamiento muy extenso, puesto que el nuevo conocimiento podría apuntar a fines comerciales o publicitarios.

---

<sup>3</sup> Base de datos semántica (DBpedia, 2014)

Para la identificación del propósito de una ontología, Mike Uschold et al. (1995) proponen la realización de preguntas de competencia para identificar de una manera más precisa el propósito de la ontología; las preguntas de comprensión se elaboran según el ámbito de aplicación basándose principalmente en el juicio experto.

La construcción de la ontología parte de lo aprendido en las preguntas de comprensión en la etapa de propósito, en este punto el diseñador puede identificar conceptos clave, relaciones y alcance del dominio de interés, el primer paso de esta etapa se conoce como captura ontológica.

**Tabla 2.** Proceso de captura en la construcción ontológica.

<b>Tarea</b>	<b>Descripción</b>
Identificación	Definición de palabras clave, relaciones, conceptos y alcance
Desambiguación	Identificación de términos similares y diferenciación
Conceptos y relaciones	Selección de los términos que se incluirán en la ontología
Validación	Se evalúa si la ontología refleja el dominio de interés.

**Fuente:** (Uschold & King, 1995).

La codificación de la ontología implica plasmar la conceptualización abstraída durante el proceso de modelamiento; se estructuran los metaconceptos que se incluirán en la metodología, luego se selecciona un lenguaje formal para concluir la etapa codificando la metodología en el formato seleccionado. Es importante diferenciar que en esta etapa no se hace referencia al proceso de codificación de software para el que la ontología esté destinada.

La integración ontológica introduce conceptos reutilizados de otras ontologías, cada lenguaje de representación tiene sus reglas de integración propia; la reutilización de conceptos puede ser total o parcial dependiendo de las necesidades del ámbito de interés, los recursos integrados pueden ser extendidos y soportar todas las relaciones de herencia descritas por OMG<sup>4</sup> en su sitio web.

---

<sup>4</sup> Ver cita (OMG, 2014).

**Tabla 3.** Proceso de Codificación en la construcción ontológica.

<b>Tarea</b>	<b>Descripción</b>
Creación de meta conceptos	La creación de estructuras que reflejen los conceptos seleccionados en la tarea de captura ontológica.
Lenguaje de representación	Selección de entre los diferentes lenguajes para la creación de ontologías como: RDF, OWL, TORTOISE
Codificación	Según las reglas del lenguaje de modelamiento seleccionado se crean las estructuras.

**Fuente:** (Uschold & King, 1995).

La evaluación responde a la necesidad de medir el grado de consistencia con la que la ontología refleja el ámbito de interés; algunas de las tareas realizadas durante el modelamiento procuran garantizar el trabajo realizado; existen muchos recursos para validar la consistencia del modelo de forma automatizada, Una de las herramientas más extendidas es Protégé<sup>5</sup>, permite probar la consistencia y se puede extender mediante la integración con diversos razonadores.

Para el proceso de documentación de la ontología Mike Uschold et al. (1995) recomiendan realizar artefactos formales que describan el proceso de creación de la ontología y publicarlos para su posterior reutilización; particularmente se menciona a Douglas Skuce (1995) quien propone algunos postulados para investigar, publicar y compartir recursos ontológicos.

#### **4.3. Metodología modular de Hois, Bhatt, & Kutz**

Uno de los puntos más críticos en el proceso de modelado ontológico es la correlación que los datos poseen, es decir que definiciones propiedades o instancias son entendidas entre ontologías; esta capacidad permite relacionar modelos dando especial atención en la modularidad y en la reutilización, de este modo se puede tener un diseño más apegado al orden real de las cosas que no solo brinde más que una descripción de los componentes y su interacción; se puede tener un modelo ontológico por cada vista<sup>6</sup> de una arquitectura con el propósito de dotarla con un valor semántico; como resultado cada vista puede conectarse e interactuar de diversas maneras, generando un modelo más consistente y modular; Joana Hois, Mehul Bhatt y Oliver Kutz (2009) proponen una metodología para la creación de ontologías orientada a la modularidad de sus componentes y demuestran su aplicación en el ámbito de la automatización de espacios arquitectónicos (Ver Anexo A).

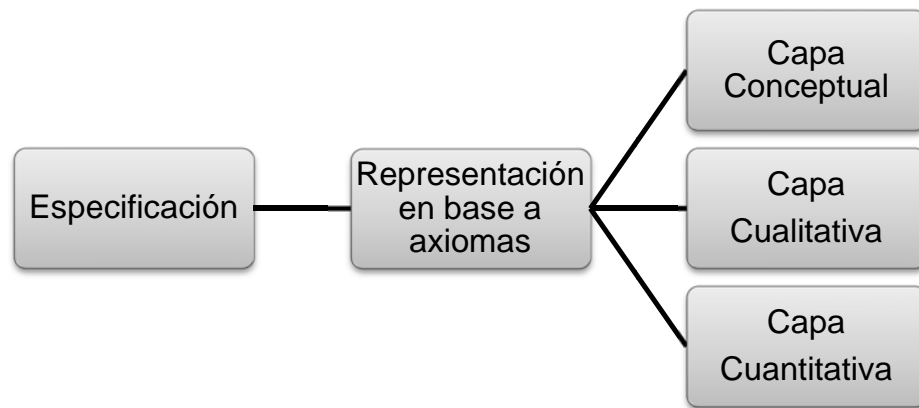
---

<sup>5</sup> Ver cita (Stanford University, 2014)

<sup>6</sup> En arquitectura de software: Artefacto que describe un ámbito particular de un componente o sistema.



La metodología de Joana Hois et al. (2009) está basada en la teoría E-conexiones<sup>7</sup>, que de manera resumida busca crear sistemas electrónicos optimizados en función del menor número de formalismos en cuanto tiene que ver a axiomas (Kutza, Lutz, Wolter, & Zakharyashev, 2004); al llevar un problema de la vida real a un escenario de automatización, el nexo entre los requerimientos del ámbito de interés y el sistema resultante está determinado por los axiomas introducidos en el proceso de modelado; según Joana Hois et al. (2009) para crear los algoritmos que respondan a una especificación, es necesario pasar primero por tres capas que definen entidades, propiedades cuantitativas y propiedades cualitativas.



**Ilustración 8.** Metodología modular (Hois, Bhatt, & Kutz, 2009).  
**Fuente:** (Hois, Bhatt, & Kutz, 2009).

La capa cuantitativa hace referencia específicamente a la información dimensional; en esta capa también sería pertinente colocar la información de las medidas mínimas y máximas de los elementos sujetos al modelado. A nivel de capa cuantitativa, también se puede adicionar información un poco más específica referente a una entidad, siempre que esté expresada en una métrica; las propiedades que no son comunes a todos los objetos y que en cierta manera los caracterizan pero no superan el plano de la capa cuantitativa deben incluirse por cada entidad.

Los axiomas generados a este nivel podrían tener menor peso al momento de resolver inconsistencias, las relaciones descritas no aportan mayor especificación sobre el comportamiento de un objeto. En la construcción de una edificación se podría realizar inferencias al moverse entre los rangos de la capa y determinar qué medidas debería tener

---

<sup>7</sup> Grado de formalismo deseable en un sistema electrónico.

una puerta; pero el razonamiento posterior de un sistema de inteligencia artificial estaría limitado a tomar decisiones muy acartonadas entre los máximos y mínimos de un rango; las inferencias realizadas si el sistema de inteligencia artificial comprende la diferencia entre una salida de emergencia y una entrada principal son mucho más valiosas, pero esta caracterización supera el nivel cuantitativo.

La capa cualitativa se centra en describir de manera formal las relaciones funcionales de las entidades de un dominio de una arquitectura, las relaciones de las entidades en esta capa están basadas en el análisis de regiones; según la metodología propuesta por Joana Hois et al. (2009) las entidades son las mismas que se definen en la capa cuantitativa; en este punto la metodología no difiere mucho de otras como por ejemplo: para Philippe Kruchten (2004) la misma entidad pasa entre capas sin cambiar más que su interpretación.

Para Joana Hois et al. (2009) al pasar de la capa cuantitativa a la capa cualitativa se está manipulando el mismo objeto, pero en esta ocasión el objeto está determinado por propiedades no métricas o aspectos funcionales de las entidades; una puerta por ejemplo en esta capa ya no estará definida por sus dimensiones sino por la propiedad funcional de ser una conexión entre habitaciones y corredores, por estar conectada o ser adyacente a otras entidades, una puerta también puede ser definida por la propiedad funcional de proveer acceso.

La capa conceptual determina el cómo una entidad puede ser diferenciada por las propiedades cualitativas determinadas por su entorno, esto facilita la toma de decisiones para aplicaciones de IA<sup>8</sup>, pero existen casos en que las propiedades diferenciadoras de una entidad no vienen dadas por el cálculo de regiones de conexión como se veía en la capa cualitativa; un objeto o un rango de objetos puede ser especializado de manera más precisa por las propiedades cualitativas reflexivas más que por las referencias de su entorno, no es una buena práctica realizar el proceso de inferencia a este nivel, ya que se incurre en la creación de reglas demasiado especializadas y redundantes; pero Joana Hois et al. (2009) recomiendan dotar a las entidades del conocimiento provisto por la capa conceptual, dado que el peso para resolver una decisión entre dominios al nivel conceptual puede servir para resolver escenarios muy particulares.

Los módulos cuantitativo, cualitativo y conceptual son integrados mediante una ontología de representación de la integración; en esta capa la teoría de las E-conexiones de Oliver Kutza et al. (2004), proporciona información adicional al proceso de construcción de componentes; en consecuencia, en la capa de representación integrada únicamente se describe la relación de las clases de diferentes ontologías y como están conectadas, pero no define restricciones

---

<sup>8</sup> Inteligencia Artificial.

respecto a los enlaces entre las ontologías; algunas ontologías pueden ser disjuntas por lo que no se puede conciliar sus entidades, esto no implica necesariamente una falla de diseño sino más bien un asunto de perspectiva.

La construcción de una representación puede moverse a través de los diferentes dominios y capas de un diseño arquitectónico, por lo que el paso de las entidades implica cumplir con algunas condiciones; las reglas que conectan las entidades a través de las capas están dadas por la capa de representación integrada, las dimensiones modulares antes descritas también pueden presentarse en este nivel; y existirán casos en los que se tenga que considerar integraciones, conexiones, extensiones definicionales, puentes (Ver Anexo A).

Si se tiene una extensión definicional de una entidad manejada por el sistema; la siguiente capa probablemente no entenderá la entidad o no será capaz de aplicarle los axiomas introducidos y referentes a la extensión definicional, esto compromete la integridad del modelo, inferencias y razonamientos que se realicen posteriormente; pero si se puede extender las formas de conectar diferentes entidades que previamente fueron introducidas en las capas anteriores; es decir, las dimensiones afectadas en esta capa afectan el cómo y no el qué ni el porqué de las entidades ya definidas.

## **5. Evaluación Metodológica.**

Para la construcción de una ontología para emociones humanas en videos de YouTube se evaluaron tres metodologías candidatas, en este apartado se explica el criterio técnico utilizado para seleccionar una metodología; se evalúan aspectos relacionados a la ontología como: usuario final, requerimientos, artefactos, escalabilidad, complejidad y recursos requeridos.

El usuario final para la ontología de emociones humanas en videos de YouTube posee un perfil muy específico, se trata de un usuario con conocimientos medios o avanzados en inteligencia artificial y manejo de recursos ontológicos; desde el punto de vista del usuario final la metodología propuesta por NeOn Proyect (2006) podría no ser la más indicada, puesto que está orientada al ámbito empresarial donde el lenguaje en el que se expresa el interesado no es técnico y por lo tanto se requiere de meticulosos procedimientos de levantamiento y licitación de requerimientos, el procedimiento es muy similar en el proceso racional unificado de Philippe Kruchten (2004); en el caso de la metodología propuesta por Mike Uschold et al. (1995) el proceso de modelado es mucho más sencillo, proponen partir de algunas preguntas básicas de comprensión para identificar entidades, conceptos, relaciones y restricciones que posteriormente serán parte de la ontología; resulta mucho más conveniente tanto para el diseñador como para el usuario final ya que implica un ahorro considerable de recursos, logrando minimizar en el costo de realización de un proyecto; de

momento se prefiere la metodología de Mike Uschold (1995) antes que NeOn Project (2006); desde el punto de vista del usuario final, al considerar la metodología propuesta por Joana Hois et al. (2009) se puede apreciar que el esfuerzo empleado para captar los requerimientos es aún menor que en el caso del marco de trabajo de Mike Uschold et al. (1995) aunque posteriormente resulta más compleja, es aún más cercana al usuario final del tipo técnico, en este contexto el mismo usuario final podría generar artefactos válidos para la creación de la ontología.

En cuanto a los requerimientos si se trabaja con NeOn Project (2006) donde el grado de confianza de cumplimiento es mayor, por la gran cantidad de recursos que la metodología emplea para poder llevar adelante el modelado, sumado al hecho de que el desarrollo ontológico está orientado a un usuario experto, NeOn Project (2006) estaría a la cabeza en el tema de los requerimientos; pero si extrapolamos lo aprendido sobre las E-conexiones de Oliver Kutza et al. (2004), lo que se quiere es buscar el menor grado de complejidad en un sistema, en este caso para producir una ontología; para la metodología propuesta por Mike Uschold et al. (1995) el proceso de levantamiento de requerimientos puede resultar muy pobre en primera instancia, pero se debe tener en cuenta que se trata de un esqueleto de metodología, esto implica que se puede reforzar la tarea con otros procedimientos para garantizar el cumplimiento de los requerimientos, del mismo modo ocurre con la metodología de Joana Hois et al. (2009) al emplear un gran número de procesos para levantar requerimientos también estaría por debajo de NeOn Project (2006). Cuando se aplica la metodología de Joana Hois et al. (2009) se trabaja con usuarios expertos, la carencia de procedimientos de levantamiento de requerimientos puede verse compensada con la experiencia, pero de igual manera en el caso de NeOn Project (2006) la utilización de consultores externos reforzaría el proceso.

Todas las metodologías generan artefactos como producto del modelado, antes de obtener el producto final, en este caso en la ontología para emociones humanas en videos de YouTube; utilizando NeOn Project (2006) se tiene un alto número de artefactos generados en comparación a las demás metodologías; la cantidad de artefactos generados depende del contexto en el que se elabora la ontología, es decir tener pocos artefactos o muchos no es un indicador por sí solo, se puede decir que al tener muchos artefactos se garantiza el cumplimiento de los requerimientos, pero la cantidad excesiva de artefactos incurre en costos, tiempo, recursos y por tanto sería mejor si no se generan demasiados artefactos; además, se está trabajando con un usuario capaz de comunicar y evaluar con un criterio muy técnico los entregables generados.

Para la metodología Mike Uschold et al. (1995) son pocos los entregables generados al igual que para Joana Hois et al. (2009); en el escenario actual resulta similar la controversia que suele generarse en el ámbito del desarrollo de software cuando se trata de entregables

generados, es común ver bandos opuestos entre RUP de Philippe Kruchten (2004) y los partidarios de Scrum (2014); para este caso específico se descarta la metodología de (NeOn Proyect (2006) por tener un alto número de artefactos; para poder discriminar entre la metodología de Mike Uschold et al. (1995) y el marco de trabajo equivalente de Joana Hois et al. (2009) se debe tomar en cuenta el valor semántico de los artefactos generados y desde esa perspectiva la mejor opción es la metodología de Joana Hois et al. (2009) .

En cuanto tiene que ver con la escalabilidad tanto NeOn Proyect (2006) como Joana Hois et al. (2009) en sus metodologías permiten realizar un modelado escalable, cuando se evalúa un escenario de aplicación para una ontología en el caso de NeOn Proyect (2006) se considera todos los recursos ontológicos y no ontológicos para el diseño, procurando la reutilización de conceptos de tal manera que se minimice el tiempo de desarrollo; para Joana Hois et al. (2009) la escalabilidad juega un rol fundamental siendo la premisa principal de la metodología, cada dimensión de una realidad ontológica posee su propia definición genérica, de tal manera que una nueva extensión definicional sea integrada en una capa específica sin alterar la forma en la que se percibe un objeto, cuando una nueva característica es introducida y supera la capa cuantitativa, involucra la introducción de axiomas relacionados para el modelamiento de un comportamiento; para Mike Uschold et al. (1995) la modularidad depende más de la abstracción del diseñador, la metodología propuesta se presenta como la secuencia más genérica de pasos para realizar una ontología, dependerá del diseñador si la implementación abarca la totalidad de un concepto o solo una parte; para describir como el software en desarrollo se adapta a un dominio, teniendo en cuenta que las ontologías de adaptación de dominio no necesariamente formarán parte del modelo ontológico final (Hoehndorf, Ngonga Ngomo, & Herre, 2009); un modelo es una abstracción de la realidad y si la apreciación del diseñador abarca un sentido amplio de un concepto, la modularidad será implícita al modelado, de lo contrario es necesario introducir mecanismos para apuntar hacia un diseño modular; finalmente la modularidad estaría más relacionada a la experiencia del diseñador que a la misma metodología que se aplique, desde el punto de vista modular se prefiere la metodología propuesta por Joana Hois et al. (2009) ya que introduce un mecanismo más formal y a la vez natural de separar un concepto en módulos, basado en apreciaciones cuantitativas, cualitativas y axiomáticas.

En términos de complejidad para el diseñador una metodología está supuesta a simplificar el proceso de desarrollo y a sistematizar las tareas implicadas, cuando no se tiene dominio de una metodología, la aplicación del proceso inherente puede resultar más complejo que un desarrollo artesanal poco consensuado; uno de los argumentos más conocidos en el ámbito del desarrollo tradicional es que el método introducido por Philippe Kruchten (2004) es demasiado complejo para el resultado que entrega y es la principal razón por la que

muchos prefieren la utilización de Scrum (2014); si se toma esta postura como base para la selección de una metodología basada en ontologías, la selección inequívoca sería la metodología introducida por Mike Uschold et al. (1995); una secuencia de pasos genérica cuya premisa es abarcar las tareas mínimas de un proceso, es sin lugar a dudas menos compleja que aplicar una metodología orientada a la satisfacción de los requerimientos como lo es NeOn Project (2006); en términos de menor complejidad para el diseñador se prefiere la metodología de Mike Uschold et al. (1995).

Uno de los puntos más riesgosos en el desarrollo de cualquier proyecto es la correcta administración de los recursos; se requiere de un determinado esfuerzo humano para el cumplimiento de las tareas de desarrollo en tiempos razonables, en el caso de NeOn Project (2006) al ser muy similar a la metodología de Philippe Kruchten (2004) requiere de un número considerable de recursos desempeñando diferentes roles; por el contrario tanto Mike Uschold et al. (1995) como Joana Hois et al. (2009) tienen un enfoque más orientado al investigador donde los equipos humanos constan de pocas personas; se prefiere la metodología de Mike Uschold et al. (1995) desde el punto de vista de recursos por requerir menor cantidad de recurso humano. Como se obtiene mejor resultado en términos de modularidad facilitando el soporte, se prefiere utilizar la metodología propuesta por Joana Hois et al. (2009).

**Tabla 4.** Evaluación metodológica.

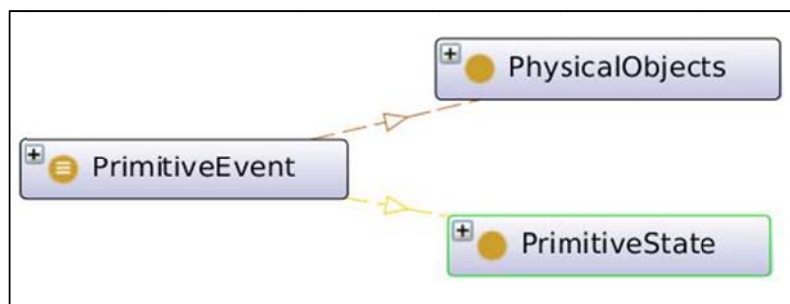
<b>Indicador</b>	<b>Metodología</b>	<b>Evaluación</b>
Tipo usuario final	(Hois, Bhatt, & Kutz, 2009)	Más apropiada para usuarios técnicos en cuanto a entregables
Requerimientos	(NeOn Project, 2006)	Optimizada para el cumplimiento de requerimientos
Artefactos	(Uschold & King, 1995)	Menor cantidad de artefactos
Escalabilidad	(Hois, Bhatt, & Kutz, 2009)	Introduce un mecanismo formal de modularidad
Complejidad	(Uschold & King, 1995)	Menor complejidad para el diseñador
Recurso Humano	(Hois, Bhatt, & Kutz, 2009)	Menor cantidad de recursos con mejores resultados

**Fuente:** Propia.

## **MODELADO ONTOLÓGICO**

## 1. Modelo conceptual

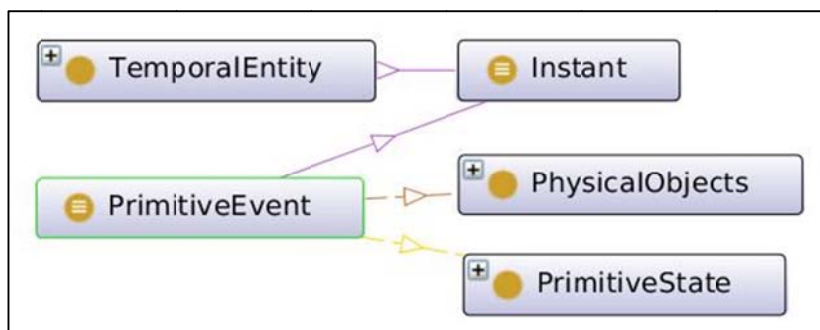
Luego de haber definido los conceptos que serán reutilizados de la ontología VERL de Francois Bremond et al. (2004) y la ontología OWL-Time (W3C, 2006); según la metodología propuesta por Joana Hois et al. (2009) es necesario definir un modelo del cual se permita especializar instancias para garantizar la modularidad del diseño; la primera capa es la definición conceptual y relaciones de las entidades que participan en la ontología, no es importante en este nivel especificar sus atributos; para la ontología de emociones humanas en videos de YouTube, la acción de apreciar una emoción humana en un video es un evento primitivo que asocia un objeto físico con estado primitivo, en este caso una persona mostrando una emoción.



**Ilustración 9.** Apreciación de eventos en videos.

**Fuente:** Propia.

Tomando en consideración que un evento primitivo no es más que la relación entre un objeto físico y un estado primitivo, falta definir el contexto temporal de la apreciación del evento, para definir las complejas relaciones temporales inmersas en el proceso de apreciación de emociones; se introdujo el concepto instante de la ontología OWL-Time (W3C, 2006), en el que un evento primitivo sucede en un instante determinado; no se considera de interés analizar eventos compuestos y solo es relevante la dimensión temporal instante, pero para garantizar la modularidad del diseño y permitir a futuro apreciar dimensiones temporales compuestas por varios instantes, se introdujo en el modelo el concepto de entidad temporal.

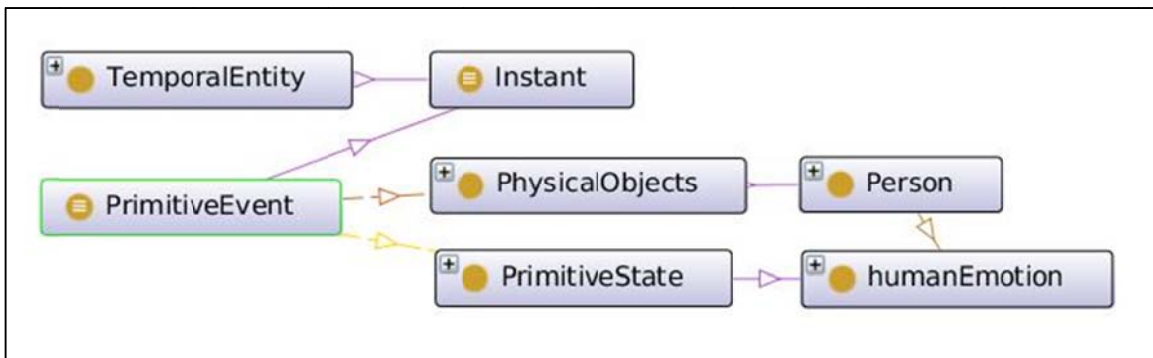


**Ilustración 10.** Relación temporal con un evento primitivo.

**Fuente:** Propia.



De manera genérica se modeló el proceso de identificación de objetos físicos en determinado estado dentro de un video, Joana Hois et al. (2009) en su metodología recomiendan extender las definiciones de las entidades para especializar el comportamiento de cada objeto apreciable; en un video se puede encontrar un persona, un árbol y un juguete, siendo los tres realidades físicas; sin embargo, un árbol tendrá un estado estático, una persona tendrá un movimiento autónomo y un juguete si es un carro a control remoto, puede tener un movimiento remotamente controlado; en este punto la herencia que se realiza del concepto objeto físico, permite asociar un comportamiento específico para cada tipo de objeto; si en un futuro se requiere reutilizar la ontología para la identificación de cualquier otro objeto diferente a una persona, solo sería necesario crear una clase que extienda de objeto físico, la acción de apreciar una entidad en un video en determinado estado permanece igual, solo cambia la clase del objeto; en este caso, el objeto físico persona mostrando el estado primitivo emoción humana.

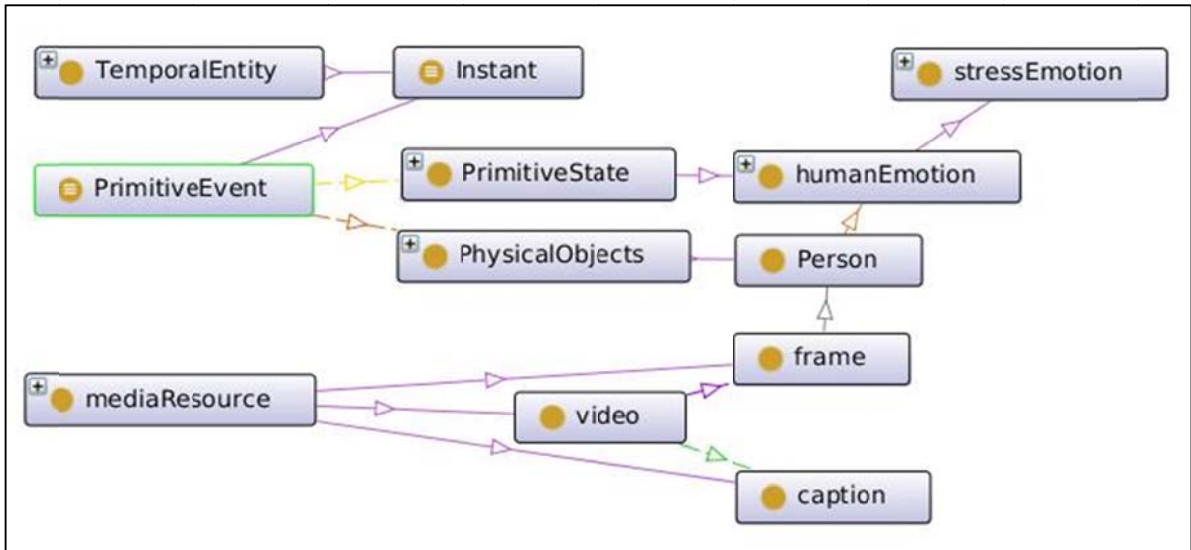


**Ilustración 11.** Especialización de evento primitivo.

**Fuente:** Propia.

Según la metodología de Jona Hois et al. (2009), es necesario introducir un concepto genérico del cual se pueda especializar el comportamiento; se introdujo el concepto de recurso multimedia del cual se especializa la clase video. Tomando en cuenta que el evento primitivo definido por Francois Bremond et al. (2004) sucede en un determinado instante de tiempo, del video se especializa otra entidad para hacer referencia al momento exacto en el que sucede la apreciación de la emoción humana; para ajustarse al modelo y relacionar el concepto video con la definición de instante de OWL-Time, se introduce el concepto de frame<sup>9</sup>, en el cual se aprecia el evento primitivo;

<sup>9</sup> Cuadro específico de una secuencia de video.



**Ilustración 12.** Modelo Emociones humanas en videos de YouTube.  
Fuente: Propia.

## 2. Capa Cualitativa

Después de haber definido las entidades que forman parte del modelo, el siguiente paso según la metodología de Joana Hois et al. (2009), fue definir la capa cualitativa de las entidades; algunas cualidades fueron tomadas de las definiciones propuestas por las ontologías reutilizadas; al definir las cualidades de un evento primitivo se introdujo componentes, para hacer referencia al estado o grupo de estados observables, se relacionó los estados a los objetos físicos apreciados en el video; se creó a nivel cualitativo una propiedad para relacionar un evento primitivo con una entidad temporal del tipo instante.

**Tabla 5.** Tripletas Evento primitivo capa cualitativa.

Sujeto	Predicado	Objeto
verl#PrimitiveEvent	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
verl#physical_objects	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
verl#componants	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
verl#ocurs_at	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty

Fuente: Propia.

En el caso de la entidad que representa un estado primitivo, solo se describe una propiedad de un objeto físico; en el modelado, un estado primitivo corresponde a una forma abstracta y según la metodología de Joana Hois et al. (2009), los atributos en el nivel cualitativo son de mayor importancia en las especializaciones; por tal motivo, para el evento primitivo solo se

requiere definir el tipo; para la clase objeto físico solo fue necesario definir el tipo a nivel cualitativo.

**Tabla 6.** Estado primitivo capa cualitativa

Sujeto	Predicado	Objeto
verl#PrimitiveState	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#Class">http://www.w3.org/2002/07/owl#Class</a>

Fuente: Propia.

**Tabla 7.** Objeto físico capa cualitativa.

Sujeto	Predicado	Objeto
verl#PhysicalObjects	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#Classes">http://www.w3.org/2002/07/owl#Classes</a>

Fuente: Propia.

Para definir el nivel cualitativo de las entidades reutilizadas de la ontología OWL-Time (2006), de donde se reutilizó los conceptos de entidad temporal e instante; fue necesario definir el atributo tipo para ambas definiciones; para el concepto instante se agregó una propiedad, que la define como una extensión del concepto entidad temporal en términos de herencia.

**Tabla 8.** Entidad Temporal capa Cualitativa.

Sujeto	Predicado	Objeto
owl-time/TemporalEntity	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#Class">http://www.w3.org/2002/07/owl#Class</a>

Fuente: Propia.

**Tabla 9.** Instante capa cualitativa.

Sujeto	Predicado	Objeto
owl-time/Instant	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#Class">http://www.w3.org/2002/07/owl#Class</a>
owl-time/Instant	<a href="http://www.w3.org/2000/01/rdf-schema#subClassOf">http://www.w3.org/2000/01/rdf-schema#subClassOf</a>	<a href="http://www.w3.org/TR/owl-time/TemporalEntity">http://www.w3.org/TR/owl-time/TemporalEntity</a>

Fuente: Propia.

Para definir el nivel cualitativo del concepto asociado a recursos multimedia, solo fue necesario definir el tipo; siguiendo los postulados de la metodología de Joana Hois et al. (2009), se definió el concepto recurso multimedia como la forma más abstracta del medio de adquisición de datos, para el dominio de emociones humanas en videos de YouTube. El concepto video es una extensión definicional de recurso multimedia, además del tipo se definió el padre. En el caso del cuadro y subtítulo a más del tipo y el padre, se definió la propiedad que relacione la apreciación del dato con la dimensión temporal; para definir de mejor manera las relaciones, se realizó una extensión definicional de la relación ya establecida entre instante y evento primitivo, adaptando cuadro y subtítulo con el momento en el video en el que son apreciados.

**Tabla 10.** Recurso multimedia capa cualitativa.

Sujeto	Predicado	Objeto
hse#mediaResource	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class

Fuente: Propia.

**Tabla 11.** Video capa cualitativa.

Sujeto	Predicado	Objeto
hse#video	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
hse#video	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://github.com/blacktiago/HSE/ontology/hse.owl#mediaResource
hse#has_a	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
hse#has_a	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	https://github.com/blacktiago/HSE/ontology/hse.owl#ocurs_at

Fuente: Propia.

**Tabla 12.** Cuadro capa cualitativa.

Sujeto	Predicado	Objeto
hse#frame	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
hse#frame	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://github.com/blacktiago/HSE/ontology/hse.owl#mediaResource
hse#show_a	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
hse#show_a	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	https://github.com/blacktiago/HSE/ontology/hse.owl#physical_objects

Fuente: Propia.

**Tabla 13.** Subtítulo capa cualitativa.

Sujeto	Predicado	Objeto
hse#caption	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
hse#caption	http://www.w3.org/2000/01/rdf-schema#subClassOf	https://github.com/blacktiago/HSE/ontology/hse.owl#mediaResource
hse#cc_at	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
hse#cc_at	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	https://github.com/blacktiago/HSE/ontology/hse.owl#ocurs_at

Fuente: Propia.

Mediante los conceptos introducidos por Francois Bremond et al. (2004) se tiene un modelo para describir el proceso de apreciación de emociones en videos; un evento primitivo muestra toda clase de realidades físicas y cada objeto físico puede tener estados propios; para completar el modelo se especializó el objeto físico y el comportamiento para el contexto de emociones en videos, se crearon extensiones definicionales para modelar el escenario específico del reconocimiento de personas; posteriormente se puede reutilizar la ontología

para el registro de cualquier objeto físico, siempre que las propiedades cualitativas introducidas derivan de las propiedades: componentes y objeto físico. Para definir una persona como objeto físico se reutilizó también las definiciones del vocabulario propuesto por FOAF (W3C, 2007) que define los atributos principales de una persona, para el reconocimiento de emociones es irrelevante la identidad de las personas que se observa, se introducen las definiciones que permiten identificar a una persona si a futuro es requerido.

**Tabla 14.** Persona capa cualitativa.

Sujeto	Predicado	Objeto
foaf:Person	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#Class">http://www.w3.org/2002/07/owl#Class</a>
foaf:Person	<a href="http://www.w3.org/2000/01/rdf-schema#subClassOf">http://www.w3.org/2000/01/rdf-schema#subClassOf</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#PhysicalObjects">https://github.com/blacktiago/HSE/ontology/hse.owl#PhysicalObjects</a>
hse#shows	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#ObjectProperty">http://www.w3.org/2002/07/owl#ObjectProperty</a>
hse#shows	<a href="http://www.w3.org/2000/01/rdf-schema#subPropertyOf">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#components">https://github.com/blacktiago/HSE/ontology/hse.owl#components</a>

Fuente: Propia.

**Tabla 15.** Cuadro capa cualitativa.

Sujeto	Predicado	Objeto
hse#show_a	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#ObjectProperty">http://www.w3.org/2002/07/owl#ObjectProperty</a>
hse#show_a	<a href="http://www.w3.org/2000/01/rdf-schema#subPropertyOf">http://www.w3.org/2000/01/rdf-schema#subPropertyOf</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#physical_objects">https://github.com/blacktiago/HSE/ontology/hse.owl#physical_objects</a>
hse#frame	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#Class">http://www.w3.org/2002/07/owl#Class</a>
hse#frame	<a href="http://www.w3.org/2000/01/rdf-schema#subClassOf">http://www.w3.org/2000/01/rdf-schema#subClassOf</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#mediaResource">https://github.com/blacktiago/HSE/ontology/hse.owl#mediaResource</a>

Fuente: Propia.

### 3. Capa Cuantitativa

El proceso de modelado ontológico propuesto por Joana Hois et al. (2009), destina la capa cuantitativa, para determinar todas las propiedades de una entidad que no aportan ningún significado semántico al modelado; las propiedades cualitativas pueden ser requeridas incluso prioritarias para diversas aplicaciones; por ejemplo, al apreciar una emoción humana en un video, se requiere conocer el tiempo exacto en horas, minutos y segundos en que se aprecia determinada emoción; las componentes numéricas del tiempo de apreciación, por si mismas no aportan ningún valor semántico sobre el proceso; sin embargo, sin el dato del cuadro exacto, no se podría consumir el recurso posteriormente.

**Tabla 16.** Cuadro capa cuantitativa.

Sujeto	Predicado	Objeto
hse#frame_at	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#DatatypeProperty
hse#frame_at	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	http://www.w3.org/TR/owl-time/inXSDDateTime

Fuente: Propia.

No todas las entidades poseen atributos de la capa cualitativa; al considerar por ejemplo las definiciones de evento primitivo, estado primitivo y objeto físico, por ser conceptos muy genéricas no superan la capa cualitativa; las extensiones definicionales introducidas para la ontología de Francois Bremond et al. (2004), por ser clases hijas reflejan comportamientos más especializados; en la capa cuantitativa se introdujeron atributos como el tiempo en el que se aprecia una emoción; aunque las definiciones de la ontología OWL-Time (2006) permiten al modelo, dar soporte a complejas realidades temporales, para la capa cuantitativa de la ontología de emociones humanas en videos de YouTube, solo es importante el tiempo en que se aprecia la emoción, en sus componentes numéricos más simples; otro dato relevante para la capa cualitativa es el subtítulo; es decir, el texto plano que contiene el subtítulo; por si solo el texto no representa ningún valor semántico, pero es necesario almacenarlo para su análisis posterior y su relación con la emoción humana apreciada en determinado instante del video.

**Tabla 17.** Subtítulo capa cuantitativa.

Sujeto	Predicado	Objeto
hse#cc_at_seconds	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#DatatypeProperty
hse#cc_at_seconds	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	http://www.w3.org/2002/07/owl#topDataProperty
hse/hse.owl#cc_at	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#DatatypeProperty
hse#cc_text	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#DatatypeProperty

Fuente: Propia.

#### 4. Capa Axiomática

Según Joana Hois et al. (2009) la última capa del modelado, es la capa axiomática que actúa sobre las capas conceptual, cualitativa y cuantitativa; en esta capa se introdujeron las restricciones para dominios y rangos, de cada una de las propiedades introducidas en las capas anteriores. Se realizó el modelado de las restricciones de cada entidad del modelo propuesto; para un evento primitivo, se definieron los predicados necesarios para hacer referencia a los estados primitivos, objetos físicos y el instante de apreciación de un evento primitivo.

**Tabla 18.** Evento primitivo capa axiomática.

<b>Sujeto</b>	<b>Predicado</b>	<b>Objeto</b>
hse#ocurs_at	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2002/07/owl#ObjectProperty">http://www.w3.org/2002/07/owl#ObjectProperty</a>
hse#ocurs_at	<a href="http://www.w3.org/2000/01/rdf-schema#range">http://www.w3.org/2000/01/rdf-schema#range</a>	<a href="http://www.w3.org/TR/owl-time/Instant">http://www.w3.org/TR/owl-time/Instant</a>
hse#ocurs_at	<a href="http://www.w3.org/2000/01/rdf-schema#domain">http://www.w3.org/2000/01/rdf-schema#domain</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent">https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent</a>
hse#physical_objects	<a href="http://www.w3.org/2000/01/rdf-schema#domain">http://www.w3.org/2000/01/rdf-schema#domain</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent">https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent</a>
verl#physical_objects	<a href="http://www.w3.org/2000/01/rdf-schema#range">http://www.w3.org/2000/01/rdf-schema#range</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl/verl#PhysicalObjects">https://github.com/blacktiago/HSE/ontology/hse.owl/verl#PhysicalObjects</a>
verl#componants	<a href="http://www.w3.org/2000/01/rdf-schema#domain">http://www.w3.org/2000/01/rdf-schema#domain</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent">https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent</a>
verl#componants	<a href="http://www.w3.org/2000/01/rdf-schema#range">http://www.w3.org/2000/01/rdf-schema#range</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveState">https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveState</a>

Fuente: Propia.

Un evento primitivo tiene un rango específico para asociar cualquier elemento físico identificable en un video; según la ontología propuesta por Francois Bremond et al. (2004); el rango definido para un evento primitivo, implica que cualquier objeto relacionado debe cumplir con la característica de ser un objeto físico; cualquier nodo que se quiera asociar a un evento primitivo debe ser una extensión definicional del concepto de objeto físico.

**Tabla 19.** Objeto físico capa axiomática.

<b>Sujeto</b>	<b>Predicado</b>	<b>Objeto</b>
verl#physical_objects	<a href="http://www.w3.org/2000/01/rdf-schema#domain">http://www.w3.org/2000/01/rdf-schema#domain</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent">https://github.com/blacktiago/HSE/ontology/hse.owl#PrimitiveEvent</a>
verl#physical_objects	<a href="http://www.w3.org/2000/01/rdf-schema#range">http://www.w3.org/2000/01/rdf-schema#range</a>	<a href="https://github.com/blacktiago/HSE/ontology/hse.owl/verl#PhysicalObjects">https://github.com/blacktiago/HSE/ontology/hse.owl/verl#PhysicalObjects</a>

Fuente: Propia.

Un video es uno de los medios de adquisición más completos de datos relevantes para emociones humanas; existen definiciones ontológicas muy completas, que poseen conceptos asociados a un video y otros recursos multimedia (W3C, 2011); sin embargo, no se reutilizaron las definiciones antes mencionadas ya que el proceso de producción, distribución y formatos no tiene ningún valor semántico en el proceso de identificación de emociones humanas.

Las restricciones definidas para el concepto video, hacen referencia a los rangos de las propiedades que relacionan un video con los conceptos cuadro y subtítulo; la capa axiomática para video, restringe también la propiedad URL a datos de tipo URI<sup>10</sup>.

<sup>10</sup> Identificador universal de recursos por sus siglas en inglés.

**Tabla 20.** Video capa axiomática.

Sujeto	Predicado	Objeto
hse#cc_at	http://www.w3.org/2000/01/rdf-schema#range	https://github.com/blacktiago/HSE/ontology/hse.owl#caption
hse#cc_at	http://www.w3.org/2000/01/rdf-schema#subPropertyOf	https://github.com/blacktiago/HSE/ontology/hse.owl#ocurs_at
hse#cc_at	http://www.w3.org/2000/01/rdf-schema#domain	https://github.com/blacktiago/HSE/ontology/hse.owl#video
hse#has_a	http://www.w3.org/2000/01/rdf-schema#range	https://github.com/blacktiago/HSE/ontology/hse.owl#frame
hse#has_a	http://www.w3.org/2000/01/rdf-schema#domain	https://github.com/blacktiago/HSE/ontology/hse.owl#video
hse#url	http://www.w3.org/2000/01/rdf-schema#range	http://www.w3.org/2001/XMLSchema#anyURI
hse#url	http://www.w3.org/2000/01/rdf-schema#domain	https://github.com/blacktiago/HSE/ontology/hse.owl#video

Fuente: Propia.

El instante específico en el que se aprecia una emoción humana, es una entidad temporal que puede ser expresada en términos de horas minutos y segundos; existe un amplio esquema para tipos de datos en ontologías (W3C, 2004); un cuadro de un video, tiene una propiedad relacionada al tiempo en el que se aprecia un evento, se definió el tipo de dato válido para el rango de la propiedad. El objeto físico del evento primitivo es una persona; como rango de la propiedad que relaciona un evento a un objeto físico, se reutiliza el concepto de persona de la ontología FOAF (W3C, 2014).

**Tabla 21.** Cuadro capa axiomática.

Sujeto	Predicado	Objeto
hse#show_a	http://www.w3.org/2000/01/rdf-schema#range	http://xmlns.com/foaf/0.1/Person
hse#show_a	http://www.w3.org/2000/01/rdf-schema#domain	https://github.com/blacktiago/HSE/ontology/hse.owl#frame
hse#frame_at	http://www.w3.org/2000/01/rdf-schema#range	http://www.w3.org/2001/XMLSchema#time
hse#frame_at	http://www.w3.org/2000/01/rdf-schema#domain	https://github.com/blacktiago/HSE/ontology/hse.owl#frame

Fuente: Propia.

Al igual que un cuadro, un subtítulo está atado a un instante específico de un video; los componentes temporales asociados al registro de una emoción, tienen por dominio el tipo de dato time (W3C, 2004); el rango para la entidad temporal ya fue definido en la capa axiomática al modelar el cuadro; existen componentes y librerías que utilizan datos de tiempo como parámetros de ejecución, en muchos casos no utilizan el formato de tiempo en horas minutos y segundos, en su lugar emplean un solo componente numérico para referirse a la cantidad de tiempo en milisegundos; para poder realizar operaciones con este tipo de componentes se introdujo una propiedad que posee como rango el tipo de dato numérico



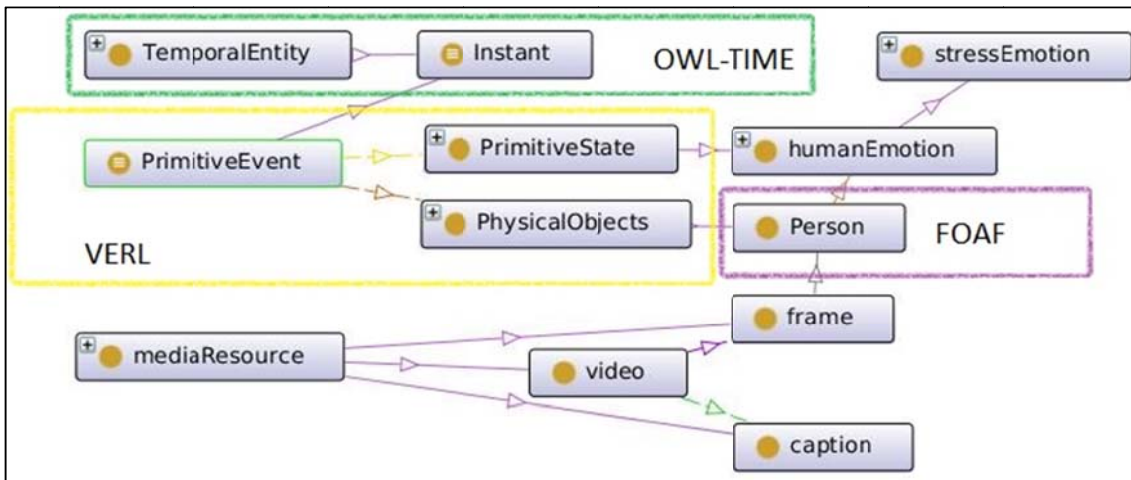
(W3C, 2004); se introdujo una propiedad para el texto del subtítulo y se definió para su rango, un tipo de dato texto (W3C, 2004).

**Tabla 22.** Subtítulo capa axiomática.

Sujeto	Predicado	Objeto
hse#cc_at_seconds	http://www.w3.org/2000/01/rdf-schema#range	http://www.w3.org/2001/XMLSchema#double
hse#cc_at_seconds	http://www.w3.org/2000/01/rdf-schema#domain	https://github.com/blacktiago/HSE/ontology/hse.owl#caption
hse#cc_text	http://www.w3.org/2000/01/rdf-schema#range	http://www.w3.org/2001/XMLSchema#string

Fuente: Propia.

Para el modelado de emociones humanas en videos de YouTube se reutilizaron 3 ontologías, se agregaron extensiones definicionales para completar el modelado; para eventos de video se reutilizó la ontología VERL propuesta por Francois Bremond et al. (2004); para manejar entidades temporales relacionadas a la apreciación de un video se reutilizó la ontología OWL-Time (W3C, 2006), dado que el estudio estuvo enfocado en rostros humanos se empleó la definición de persona de la ontología FOAF (W3C, 2007).



**Ilustración 13.** Reutilización ontológica

Fuente: Propia.

## 5. Resultado del Modelado Ontológico

Luego de haber definido un modelo que refleja el proceso de observación de emociones humanas en videos de YouTube, fue necesario agregar instancias de emociones humanas; por la gran diversidad de emociones que pueden ser apreciadas en una persona, la tarea de crear instancias de emociones requiere un sustento teórico; W. Gerrot Parrot et al. (1991), proponen una matriz de emociones bastante completa categorizando las emociones en tres niveles; cada nivel está estructurado en varias jerarquías, que relacionan las emociones de

las más genéricas a las más especializadas y complejas; el modelo introducido por W. Gerrot Parrot et al. (1991), proponen 6 emociones primarias: amor, alegría, sorpresa, ira, tristeza y miedo; los niveles 2 y 3 de la matriz agrupan varias emociones que pueden ser consideradas, como formas muy especializadas de las emociones primarias; el conocimiento aportado por W. Gerrot Parrot et al. (1991), permite introducir una estructura de emociones, para realizar inferencias que pueden incluso anticiparse a eventos emocionales; es decir, si se entiende el nerviosismo como una forma de miedo y la ansiedad como una forma de nerviosismo, se podrían crear controles preventivos mucho más efectivos en sistemas de seguridad; si se limita la detección de emociones a componentes extremos como terror o ansiedad, el control aplicado puede ser inefectivo, tardío o hasta inútil, pero si se pueden establecer patrones de conducta consecuentes para sistemas de vigilancia, el resultado reducirá el tiempo de respuesta y anticipará toda clase de incidencias de seguridad.

Primary emotion	Secondary emotion	Tertiary emotions
Love	Affection	Adoration, affection, love, fondness, liking, attraction, caring, tenderness, compassion, sentimentality
	Lust	Arousal, desire, lust, passion, infatuation
	Longing	Longing
Joy	Cheerfulness	Amusement, bliss, cheerfulness, gaiety, glee, jolliness, joviality, joy, delight, enjoyment, gladness, happiness, jubilation, elation, satisfaction, ecstasy, euphoria
	Zest	Enthusiasm, zeal, zest, excitement, thrill, exhilaration
	Contentment	Contentment, pleasure
	Pride	Pride, triumph
	Optimism	Eagerness, hope, optimism
	Enthrallment	Enthrallment, rapture
	Relief	Relief
Surprise	Surprise	Amazement, surprise, astonishment
Anger	Irritation	Aggravation, irritation, agitation, annoyance, grouchiness, grumpiness
	Exasperation	Exasperation, frustration
	Rage	Anger, rage, outrage, fury, wrath, hostility, ferocity, bitterness, hate, loathing, scorn, spite, vengefulness, dislike, resentment
	Disgust	Disgust, revulsion, contempt
	Envy	Envy, jealousy
	Torment	Torment
Sadness	Suffering	Agony, suffering, hurt, anguish
	Sadness	Depression, despair, hopelessness, gloom, glumness, sadness, unhappiness, grief, sorrow, woe, misery, melancholy
	Disappointment	Dismay, disappointment, displeasure
	Shame	Guilt, shame, regret, remorse
	Neglect	Alienation, isolation, neglect, loneliness, rejection, homesickness, defeat, dejection, insecurity, embarrassment, humiliation, insult
	Sympathy	Pity, sympathy
Fear	Horror	Alarm, shock, fear, fright, horror, terror, panic, hysteria, mortification
	Nervousness	Anxiety, nervousness, tenseness, uneasiness, apprehension, worry, distress, dread

**Ilustración 14.** Matriz de emociones humanas (Parrott & Schulkinb, 1991).

**Fuente:** (Handel, 2011).

Para aportar mayor valor semántico a la ontología, se vinculó las instancias de emociones humanas a definiciones ontológicas existentes en la web (DBpedia, 2014); existen definiciones para casi todas las emociones de la matriz propuesta por W. Gerrot Parrot et al. (1991); para ayudar a definir de mejor manera una emoción, se introdujeron predicados que complementan la definición de las emociones; en el Anexo D se muestra una tabla con propiedades, que ayudan a relacionar la matriz emocional de W. Gerrot Parrot et al. (1991) con definiciones ontológicas en la web. En la capa cualitativa se determina de donde

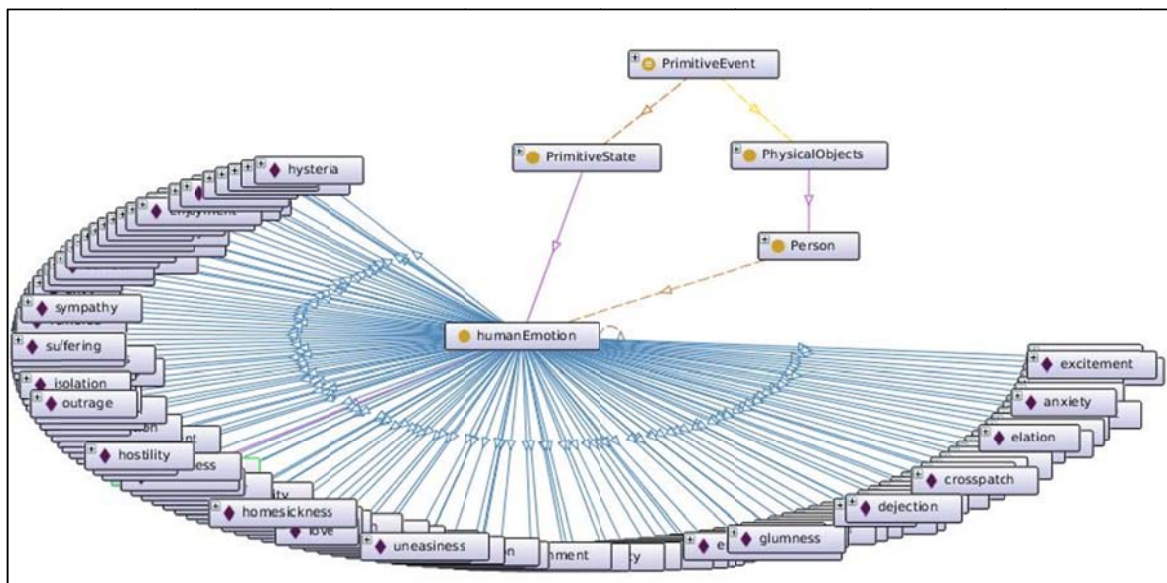
consumir la definición de la emoción, el nivel y el padre dentro de la matriz de W. Gerrot Parrot et al. (1991); si una emoción no posee padre implica que corresponden al nivel primario de la matriz.

**Tabla 23.** Emoción humana (Parrott & Schulkinb, 1991) y (DBpedia, 2014)

Sujeto	Predicado	Objeto
hse#parrott_level	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#DatatypeProperty
hse#parrotts_Child_Of	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#ObjectProperty
hse#definition	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#DatatypeProperty

Fuente: Propia.

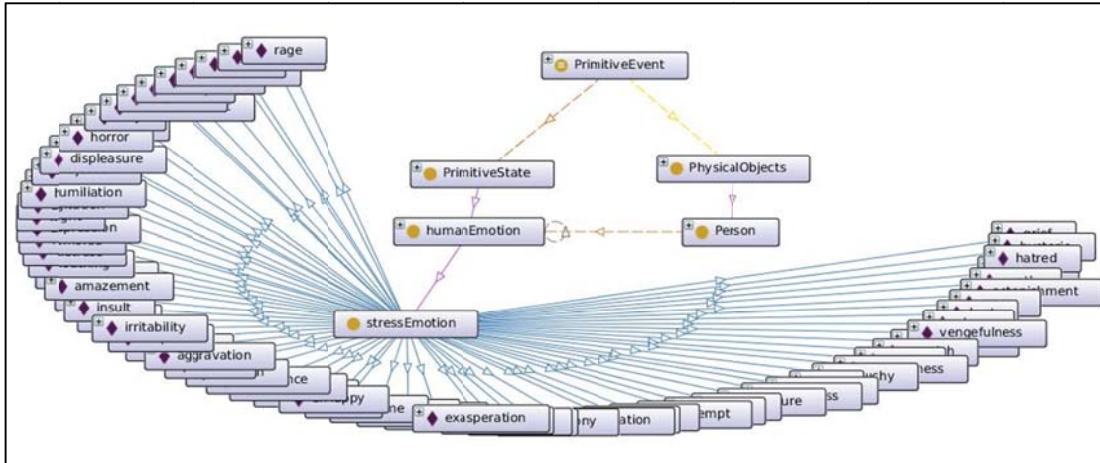
Se realizó un proceso manual de instanciación de 144 nodos, que dotan a la ontología de las emociones que podrían identificarse en un video; aun usando el esquema de W. Gerrot Parrot et al. (1991), fue necesario definir un criterio para extraer emociones de estrés humano, que implicarían una respuesta inmediata por un agente inteligente, una red ontológica o un sistema de seguridad avanzado.



**Ilustración 15.** Instancias de emociones humanas.

Fuente: Propia.

Para diferenciar las emociones que pueden ser objeto de alerta en un sistema de seguridad, se tomaron las tres últimas emociones primarias, del modelo de W. Gerrot Parrot et al. (1991) y sus jerarquías internas; el nuevo predicado se definió como una extensión definicional de emoción humana; las emociones: ira, tristeza y miedo se consideran emociones de estrés humano, por producir reacciones extremas en una persona y por lo tanto son de mayor interés para prevenir eventos no deseados en sistemas de vigilancia.



**Ilustración 16.** Emociones de estrés humano.

Fuente: Propia.

Todas las emociones humanas contenidas por la matriz de W. Gerrot Parrot et al. (1991), fueron incorporadas en la ontología. Con la extensión definicional para emociones de estrés, para casos en los que no se pudo encontrar una definición ontológica, se puede recurrir a la definición del padre; por ejemplo: no se tiene definición ontológica para horror, pero se considera que el terror es una forma especializada del miedo.

**Tabla 24.** Instancias de Miedo según W. Gerrod Parrott (1991) con definiciones de DBpedia (2014)

Fear	Horror	Fear	<a href="http://dbpedia.org/resource/Fear">http://dbpedia.org/resource/Fear</a>	Miedo
				Horror
		Alarm		Alarma/Susto
		Shock		Escandalizar
		Fear	<a href="http://dbpedia.org/resource/Fear">http://dbpedia.org/resource/Fear</a>	Miedo
		Fright		Susto
		Horror		Horror
		Terror	<a href="http://dbpedia.org/resource/Terror">http://dbpedia.org/resource/Terror</a>	Terror
		Panic	<a href="http://dbpedia.org/resource/Panic">http://dbpedia.org/resource/Panic</a>	Pánico
		Hysteria	<a href="http://dbpedia.org/resource/Hysteria">http://dbpedia.org/resource/Hysteria</a>	Histeria
	Mortification		Mortificación	
	Nervousness		<a href="http://dbpedia.org/resource/Nervousness">http://dbpedia.org/resource/Nervousness</a>	Nerviosismo
		Anxiety	<a href="http://dbpedia.org/resource/Anxiety">http://dbpedia.org/resource/Anxiety</a>	Ansiedad
		Suspense	<a href="http://dbpedia.org/resource/Suspense">http://dbpedia.org/resource/Suspense</a>	Incertidumbre/Suspense
		Uneasiness		Inquietud
		Apprehension	<a href="http://dbpedia.org/resource/Apprehension">http://dbpedia.org/resource/Apprehension</a>	Aprensión
		Worry	<a href="http://dbpedia.org/resource/Worry">http://dbpedia.org/resource/Worry</a>	Preocupación
		Distress	<a href="http://dbpedia.org/resource/Distress">http://dbpedia.org/resource/Distress</a>	Angustia/Dolor
		Dread		Pavor

Fuente: Propia.

## **APLICACIÓN DE MANTENIMIENTO ONTOLÓGICO**

## 1. Proceso General y Arquitectura

### 1.1. Software Ontológico en Monitoreo y Control

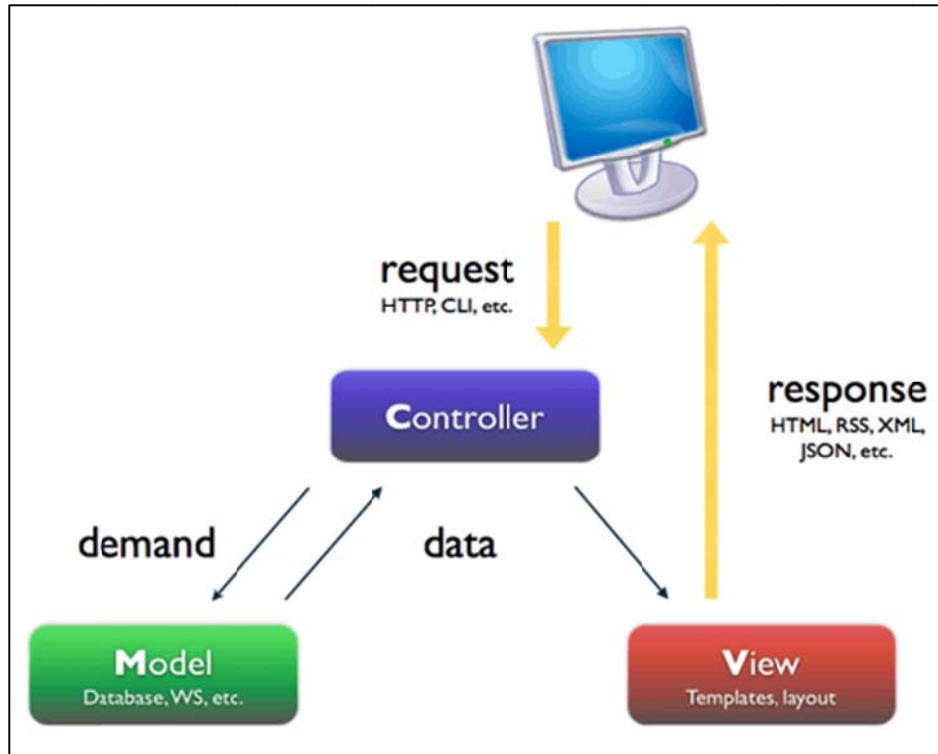
El desarrollo de sistemas de monitoreo y seguridad ha tenido un marcado repunte en los últimos años; comenzó con cámaras de muy baja resolución, que continuamente eran monitoreadas por una persona encargada de la seguridad; cuando un operador detectaba una anomalía, se procedía a tomar determinada acción; estos sistemas demostraron tener muchos puntos de falla, como prueba de ello fueron vulnerados en múltiples ocasiones; una gama más avanzada que integran múltiples sensores en sistemas distribuidos, con la capacidad de detectar anomalías o comportamientos sospechosos, mejoraron la tarea de reconocimiento. En ámbitos de AI<sup>11</sup> se dispone de un grupo de sistemas electrónicos, con diversos tipos de sensores, que reaccionan a la presencia de las personas bajo determinadas condiciones; José Castillo et al. (2011), en su investigación mencionan la necesidad urgente de crear un marco de trabajo, que agrupe metodologías, patrones de diseño y mejores prácticas para la generación de aplicaciones de vigilancia, monitoreo y control.

En lo que tiene que ver a sistemas de monitoreo y control José Castillo et al. (2011), menciona que existe una gran tendencia en el desarrollo de aplicaciones de segmentación, seguimiento de personas y reconocimiento de actividad humana, cubriendo grandes áreas físicas con toda clase de sensores; todo el volumen de datos generados, requieren de las bondades de la computación distribuida y en virtud de ello, se diseñan sistemas con diversos nodos de procesamiento con propósitos específicos; esta forma de concebir la arquitectura brinda escalabilidad y robustez en el diseño, construcción y mantenimiento de sistemas; uno de los detalles más interesantes de este tipo de sistemas, es que los nodos realizan procesamiento de los datos en tiempo real, para lanzar advertencia al operador humano más próximo, tras detectar determinados eventos predefinidos en el sistema. Algunas de las tareas más complejas, involucran el manejo de información sensorial de diversas fuentes, restricciones de la red de conciencia, comunicación entre nodos y el aprendizaje automático; una compleja estructura física de elementos de seguridad como cámaras, sensores y monitores, requieren un sistema que de soporte para la interacción entre los componentes, José Castillo et al. (2011) en su investigación, proponen un marco de trabajo para la creación de aplicaciones orientadas a redes ontológicas modulares; la idea es utilizar una adaptación de un patrón descrito en la de arquitectura de software conocido como modelo vista controlador (Reenskaug, 2003) descrito en la Ilustración 17. Modelo vista controlador.

---

<sup>11</sup> Inteligencia de ambientes por sus siglas en Inglés.





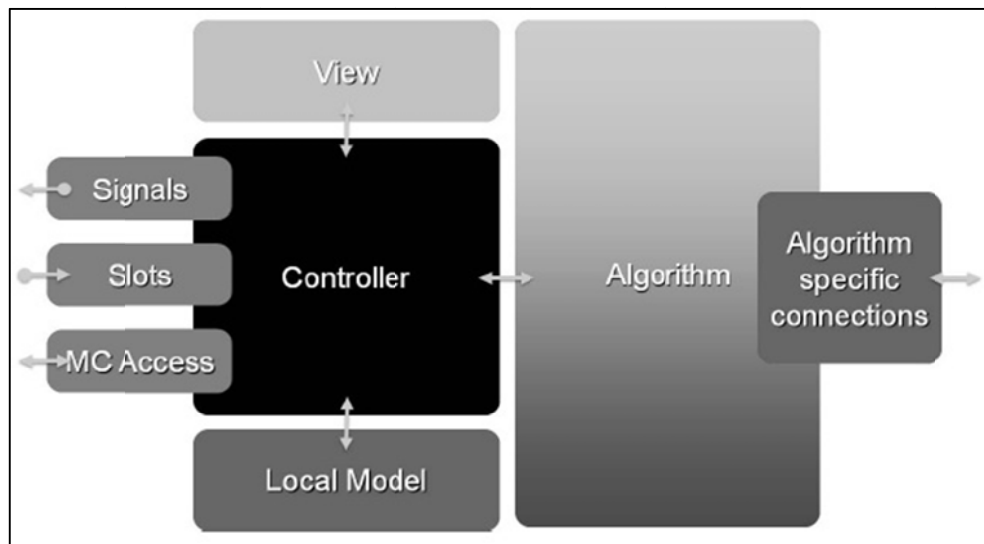
**Ilustración 17.** Modelo vista controlador.

**Fuente:** (Potencier, 2014).

En la concepción tradicional del modelo vista controlador, el modelo contiene las definiciones de todas las entidades en la aplicación, se encarga de la persistencia de los datos hacia una base de datos, un archivo, servicios web, almacenamiento en la nube, etc. El proceso de introducir, eliminar, consultar o editar datos de un medio de almacenamiento es transparente al usuario ya que por razones de seguridad y consistencia, el usuario no realiza operaciones directas sobre la base de datos; para validar las operaciones y datos ingresados por el usuario, el modelo propuesto por Trygve Reenskaug (2003), introduce el componente controlador para encapsular el modelo; el controlador se encarga de los detalles de conexión, acceso, validación de los datos hacia el usuario, etc. Cuando se realiza una operación sobre una entidad en un sistema, el controlador recibe los datos del usuario y los valida; en algunos casos, el controlador completa los datos para luego persistirlos; aunque el controlador posee todas las implementaciones para dar mantenimiento a los datos, es necesarios presentarlos de tal manera que el usuario pueda entenderlos y modificarlos; para poder entregar un formato entendible al usuario, surge el componente vista, que son todos los elementos visuales de cara al usuario, los datos entregados por el controlador, son introducidos en la vista para que el usuario pueda trabajarlos, una vista puede ser un aplicativo de consola, interfaz de escritorio, página HTML, un aplicativo móvil o cualquier interfaz a la que acceda un usuario.



El modelo propuesto por Trygve Reenskaug (2003), es uno de los más usados en el área del desarrollo de software; José Castillo et al. (2011), proponen agregar un nuevo componente al esquema tradicional del modelo vista controlador, agrupando los axiomas necesarios, para el procesamiento distribuido de los datos del sistema; la capa introducida para extender el modelo se conoce como algoritmo; la arquitectura modificada puede apreciarse en la Ilustración 18. Extensión MVC.



**Ilustración 18.** Extensión MVC.

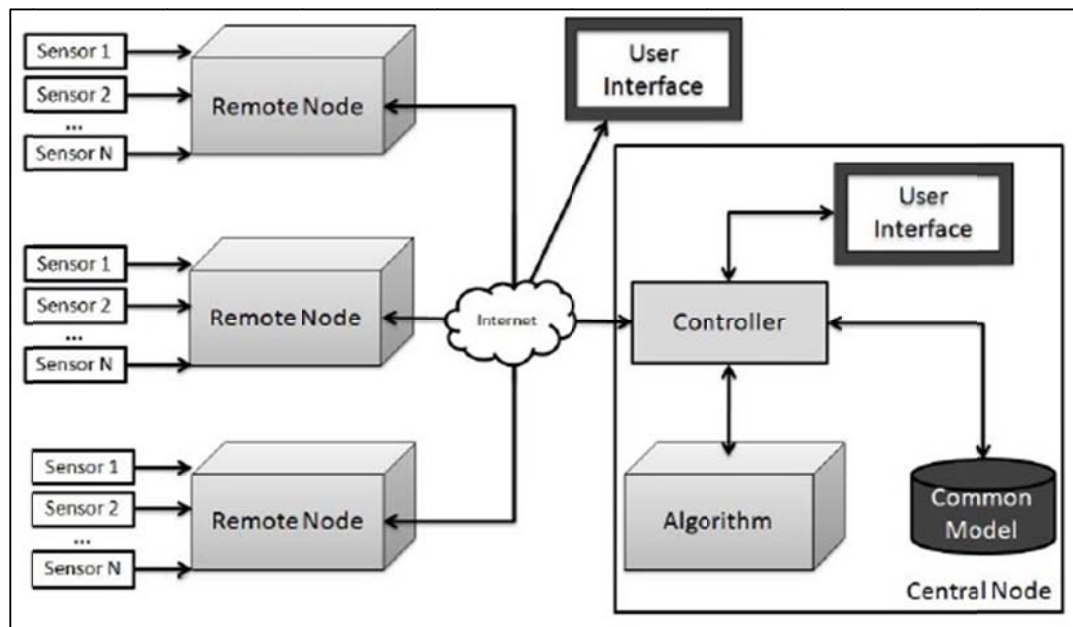
**Fuente:** (Castillo, Fernández, & López, 2011).

La capa algoritmo está orientada a trabajar a través del controlador; en un sistema distribuido un número  $n$  de nodos que cumplen diversas tareas, el controlador puede consumir información de fuentes externas al usuario mediante puertos, sensores y servicios; los datos ingresados por el cliente, no necesariamente deben ser entregados al controlador a través de una vista; en un sistema de seguridad de un supermercado, se podría tener información referente al monitoreo de una cámara de seguridad, sensores de movimiento, detección de metales o armas, identificación de rostros e incluso antecedentes penales; para evaluar de manera integral un escenario, todas las entradas deben ser consideradas por el sistema, para evaluar las circunstancias y determinar una respuesta; José Castillo et al. (2011) mencionan en su investigación, que la introducción de la capa algoritmo le brinda al sistema un mecanismo más flexible y escalable, para dar mantenimiento a los algoritmos existentes e introducir nuevos; en función del tipo de sensor, el componente algoritmo es el encargado de resolver el razonamiento que debe hacerse sobre los datos de entrada; no está directamente relacionado a los medios de adquisición de datos, trabaja a alto nivel con las entidades que comprende todo el sistema. Para que el componente algoritmo pueda realizar un proceso de inferencia apropiado, se requiere que los datos obtenidos en cada

medio de adquisición, sean procesados previamente en el controlador, antes de que el sistema resuelva la respuesta a determinado comportamiento.

## 1.2. Diseño del sistema

En el modelo propuesto por José Castillo et al. (2011), se utiliza un modelo vista controlador extendido, para introducir fuentes de adquisición de datos diferentes a una interfaz gráfica de usuarios; el controlador recibe los datos y mediante el componente algoritmo, evalúa el escenario para determinar las salidas o alertas que el sistema deba emitir al usuario. La aplicación de emociones humanas en videos de YouTube, se desarrolló para ser parte de un sistema distribuido, capaz de emitir alertas para sistemas de seguridad en entornos inteligentes; el propósito es hacer un acercamiento a una base de conocimiento, que permita analizar rostros humanos y conversaciones, para evaluar qué tipo de emoción está experimentando la persona; el aplicativo desarrollado estará encargado de mejorar la manera en que el sistema evalúa una emoción; José Castillo et al. (2011), proponen una arquitectura donde los nodos de adquisición de datos, se comunican con un nodo central mediante internet como se muestra en la Ilustración 19. Sistema distribuido sensores múltiples.



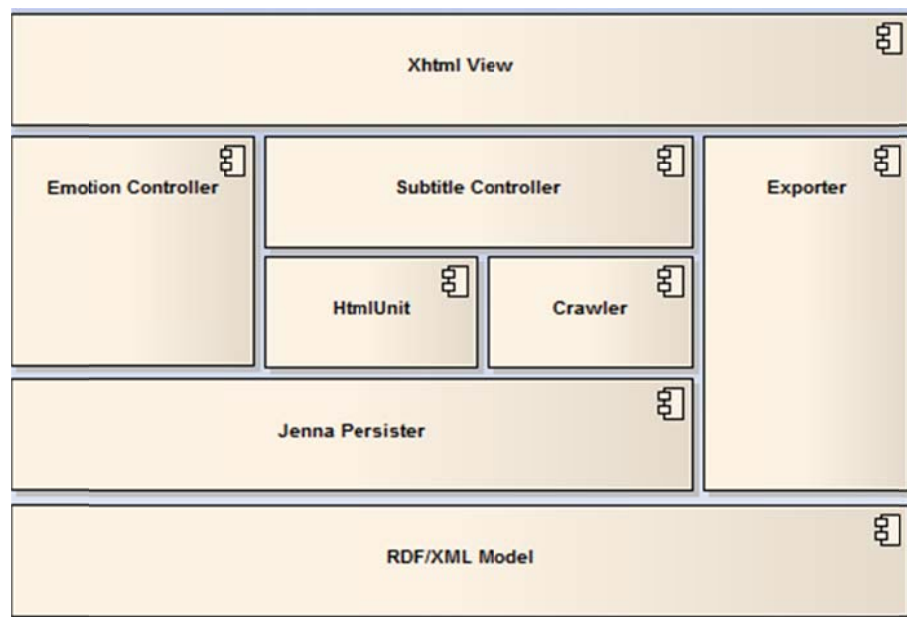
**Ilustración 19.** Sistema distribuido sensores múltiples.

**Fuente:** (Castillo, Fernández, & López, 2011).

El aplicativo desarrollado para mantenimiento de emociones humanas en videos de YouTube, podrá ser integrado posteriormente a un sistema de múltiples sensores utilizando la arquitectura propuesta por José Castillo et al. (2011); el aplicativo entregará al sistema una ontología con instancias de emociones y subtítulos en determinados instantes de un

video, para que un nodo central procese la ontología y complemente el componente algoritmo; la ontología de emociones humanas permitirá identificar patrones respecto a la relación entre lo que la persona dice y la emoción que experimenta; en resumen, el propósito del nodo que se desarrolló es que el sistema pueda mejorar la efectividad del reconocimiento, conforme un operador encuentra más instancias de emociones en videos de YouTube, se realizan operaciones de inserción, borrado, actualizado y consulta sobre un fichero RDF/XML; se utilizó la librería apache Jena (The Apache Software Foundation, 2014), para dar soporte a sentencias SPARQL, que permiten el mantenimiento de los datos del archivo.

Para obtener información y visualizar los videos en la herramienta se utilizó el API<sup>12</sup> de YouTube (2014); para la extracción de subtítulos, se utilizó un navegador virtual y un crawler<sup>13</sup> de Gargoyle Software (2014); se creó un componente que hace uso de la librería de Jena, para exportar datos de las diferentes emociones en diversos formatos; finalmente se realizaron las vistas en formato XHTML del aplicativo, utilizando el marco de trabajo de desarrollo web JSF 2.0<sup>14</sup> de Oracle (2014), para crear una interfaz web que permita manipular el fichero RDF/XML; en la Ilustración 20. Componentes aplicación de emociones humanas en videos de YouTube., se puede ver de manera general los componentes del aplicativo desarrollado.



**Ilustración 20.** Componentes aplicación de emociones humanas en videos de YouTube.  
**Fuente:** Propia.

<sup>12</sup> Interfaz de Programación de Aplicaciones

<sup>13</sup> Técnica de extracción automática de información de páginas web.

<sup>14</sup> Java Server Faces 2.0

## **2. Herramientas**

### **2.1. Jena**

Jena es un marco de trabajo escrito en Java para construir aplicaciones de web semántica y datos enlazados, fue creado por The Apache Software Foundation (2014). En desarrollo de software tradicional, es muy común la utilización de bases de datos para el almacenamiento de la información que maneja una aplicación; las aplicaciones ontológicas no son diferentes, para el almacenamiento de ontologías, se utilizan bases de datos gráficas como Virtuoso de OpenLink (2014) o 4store de Garlik (2009); las bases de datos gráficas poseen librerías y exponen servicios, que permiten realizar el mantenimiento de los datos mediante sentencias SPARQL; por lo general se realizan solo consultas sobre las ontologías y las bases de datos gráficas no se explotan de manera transaccional; Michael Grobe (2009), demuestra como pasar de una estructura de predicados, a un archivo RDF/XML mediante el uso de Jena; también expone cómo estructurar consultas para consumir recursos ontológicos. Bases gráficas como virtuoso poseen librerías que permiten a los desarrolladores conectarse a las bases de datos y realizar operaciones sobre los datos almacenados.

### **2.2. Api de YouTube**

El aplicativo de mantenimiento de emociones humanas en videos de YouTube, se realizó con vistas XHTML según las especificaciones JSF 2.0 (Oracle, 2014); en el proceso de registro y mantenimiento de los datos, se requería poder visualizar los videos en la herramienta, para que un operador pudiera detener la reproducción del video y registrar el momento en el que se aprecia la emoción; se hizo uso de las librerías proporcionadas por YouTube (2014); luego de registrar los datos, se procedió a guardarlos en el archivo RDF/XML mediante Jena, con un componente desarrollado para la persistencia de las entidades; se desarrolló una librería de JavaScript, para poder interactuar con el api de YouTube y realizar operaciones como pausar, cargar un video, extraer el tiempo en el que un operador aprecia una emoción y enviar los datos al controlador, para que se encargue de realizar operaciones sobre el modelo ontológico; se presentó una pantalla que permite visualizar los videos por emoción y otra que permite registrar varias emociones por video.

### **2.3. Extracción de subtítulos**

Para realizar inferencias más precisas sobre las emociones registradas en los videos de YouTube, se requiere extraer en formato de texto plano, los subtítulos de los videos cuando se encuentran disponibles; con la información de los subtítulos se puede realizar análisis de sentimiento, para completar la información que permitirá a un sistema de redes ontológicas identificar una emoción, en función del rostro registrada y lo que la persona está diciendo; Google (2014) en su sitio web para desarrolladores, expone métodos que permiten consumir los subtítulos de un video en diferentes formatos, pero solo permite consumir la información

de videos de los que el usuario es propietario; con la restricción del API de YouTube, se pierde el vasto conocimiento que existe en los videos ya subidos, además que se requiere realizar operaciones de autenticación mediante una cuenta de correo de Google, para poder consumir el servicio de subtítulos. Para tener información de cualquier video de YouTube sin requerir de una cuenta de Google, inicialmente se planteó descargar el video y utilizar librerías de JMF<sup>15</sup> de Oracle (2014), para separar el audio del video; con el audio obtenido se planteó utilizar las librerías de Google Chrome Apps (2014), para transformar el audio a texto; se rechazó la idea del uso de del API de Google para procesamiento de voz, para evitar la dependencia del navegador de Google; además, la solución propuesta implicaba demasiado tiempo de procesamiento para reproducir el audio, mientras el API de reconocimiento de voz de Google, transforma el audio palabra por palabra; el tiempo de procesamiento sería directamente proporcional a la duración del video; si un video tiene una duración de una hora, el aplicativo tardaría una hora en procesar el audio; en aplicaciones web el tiempo de espera de un usuario esta dado en el orden de los segundos, una hora de espera es inaceptable. Luego de descartar el uso del API de reconocimiento de voz de Google, se planteó la implementación de un procesador de voz personalizado, basado en las librerías CMU-SPHINX (Carnegie Mellon University, 2014), la idea también fue descartada por implicar demasiado tiempo de procesamiento. Otra alternativa para la extracción de subtítulos, fue crear una implementación de la librería de reconocimiento de voz, originalmente creada por Sun Microsystems (2008); para poder realizar la implementación se requería el uso de una gramática mediante las especificaciones del lenguaje JSpeech Grammar Format (W3C, 2000); la idea de crear un procesador de voz resultaba demasiado compleja, por la estructuración de la gramática necesaria para dar soporte al reconocimiento de voz y poder transformar el audio de los videos en texto plano. Finalmente se optó por utilizar los mismos subtítulos que provee YouTube para la mayoría de los videos; para poder extraer los subtítulos de cualquier video sin la necesidad de poseer una cuenta de correo de Google, se decidió utilizar técnicas de virtualización y Crawling<sup>16</sup> para la extracción de los subtítulos de manera automatizada; la mayor ventaja de utilizar esta técnica, es la reducción considerable del tiempo de ejecución en la herramienta web; el tiempo de extracción de los subtítulos, depende del ancho de banda de la conexión y es equivalente al tiempo que emplea el usuario en cargar un video, disparar el evento de carga de subtítulos y extraerlo de la página web.

---

<sup>15</sup> Marco de trabajo multimedia de Java por sus siglas en Inglés.

<sup>16</sup> Programa para extraer información de páginas web también conocido como araña.

**Tabla 25.** Tiempos de extracción de subtítulos por tecnología.

<b>Tecnologías</b>	<b>Descripción</b>	<b>Proporcional a</b>
JMF Google Api	Extracción de canal de audio y transformación de voz a texto.	Duración del video
JMF JSAPI JSGF	Extracción de canal de audio y transformación de voz a texto.	Duración del video
JMF CMU-Sphixs	Extracción de canal de audio y transformación de voz a texto.	Duración del video
HTMLUnit Crawler	Carga y manipulación virtual y extracción de HTML.	Ancho de banda

Fuente: Propia.

Tras evaluar las diferentes tecnologías para extraer el texto correspondiente a lo que las personas dicen en los videos de YouTube, se optó por utilizar la librería htmlUnit<sup>17</sup> de Gargoyle (2014), que permite crear instancias virtuales de navegadores en memoria y manipular los elementos HTML de una página, para lanzar la acción sobre el elemento que carga los subtítulos y transformar los elementos HTML a un objeto Java que puede ser persistido en el archivo RDF/XML. El aplicativo de mantenimiento de emociones humanas en YouTube, presenta los subtítulos por video; cuando no existen subtítulos en el recurso ontológico la unidad HTML carga el video en el navegador virtual, genera el evento de carga de subtítulos y extrae los datos; para controlar la extracción de subtítulos mediante el aplicativo, se generaron vistas XHTML para interactuar con el operador.

### **3. Desarrollo**

#### **3.1. Marco de trabajo de desarrollo y redes ontológicas**

Existen diversas metodologías para el desarrollo de software que responden a diversos enfoques como: requerimientos, tiempo de desarrollo, calidad, etc. Cada una de estas metodologías agrupa una serie de disciplinas que involucran la recolección de información, análisis, diagramas y muchos otros artefactos que describen el proceso realizado.

Uno de los marcos de trabajo más usado para el desarrollo de proyectos de software es conocido como proceso racional unificado (Kruchten, 2004), describiendo una serie de especificaciones y modelos basados en UML(OMG, 2014); RUP<sup>18</sup> es catalogado como una

---

<sup>17</sup> Herramienta de prueba de aplicaciones web que permite emular navegador de manera virtual.

<sup>18</sup> Proceso Racional Unificado.

metodología formal por tener un alto nivel de abstracción que alcanza y a la complejidad que puede llegar a tener, aunque también es importante tener claro que es flexible y permite versiones más ligeras según las necesidades del proyecto en desarrollo. Otra opción en cuanto a metodología de desarrollo de software es SCRUM<sup>19</sup>, que pretende realizar proyectos en el menor tiempo posible y por tal motivo se la conoce como una metodología ágil (Scrum, 2014); un grado de complejidad bajo y el hecho de que la cantidad de artefactos generados es menor que en RUP, posicionan a SCRUM como una filosofía opuesta hasta cierto punto a las metodologías tradicionales.

Las metodologías de desarrollo de software responden a la necesidad de llevar problemas de la vida real a entornos informatizados; en este punto surge la diferencia entre usar metodologías tradicionales y emplear metodologías ontológicas en la creación de soluciones informáticas; no es lo mismo tener que dar soporte a los problemas de la vida real que tener que describir los problemas de la vida real. Si se piensa en entornos de automatización donde no se puede modelar casos de uso e historias de usuario para todos los posibles flujos que se pueden dar en un escenario, resulta conveniente un razonable número de reglas que lleven a un sistema a comportarse de una u otra manera, al punto de generar respuestas apropiadas a casos que el diseñador ni siquiera ha considerado, pero que un grupo de axiomas puede llevar a discernir el comportamiento deseado.

### 3.2. Sentencias SPARQL

El aplicativo para mantenimiento de emociones de estrés humano en YouTube, trabaja directamente sobre un archivo RDF/XML, que será incorporado a la capa algoritmo de un modelo de la arquitectura propuesta por José Castillo et al. (2011); fue necesario realizar operaciones con sentencias SPARQL (W3C, 2008) a través de la librería Jena, para dar soporte a las operaciones de consulta, inserción, edición y borrado de datos.

```
PREFIX vm: <https://github.com/blacktiago/HSE/ontology/hse.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT *
  WHERE {
    ?emotion rdf:type vm:stressEmotion . };
```

**Ilustración 21.** Consulta SPARQL para obtener emociones de estrés humano.

**Fuente:** Propia.

Para agregar nodos en una ontología que correspondan a instancias de una clase o a individuos, al igual que en una base de datos relacional se requiere un identificador; cuando

---

<sup>19</sup> Metodología de desarrollo ágil descrita en <https://www.scrum.org/>.

se trabaja siguiendo los principios de la web semántica como lo indican Tim Berners-Lee et al. (2009), se generan identificadores de tipo URI para cada recurso; no es recomendable usar identificadores basados en el ingreso secuencial de los datos en la ontología; la razón principal de generar identificadores en función de nodos, es optimizar las búsquedas y modelar de manera más precisa el orden real de las cosas; para introducir un individuo en un recurso ontológico; la especificación del lenguaje SPARQL (2008), recomienda estructurar una sentencia indicando que un identificador debe ser generado; mediante el uso de los caracteres guion bajo seguido de dos puntos, las bases de datos grafica como virtuoso de OpenLink (2014), interpretan la sentencia y generan el identificador según una secuencia lógica antes que una ordinal; en la ontología de emociones de estrés humano en videos de YouTube, se estructuró una sentencia SPARQL que permite guardar los datos con identificadores generados automáticamente por Jena.

```
PREFIX vm: <https://github.com/blacktiago/HSE/ontology/hse.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/spec/#>
INSERT DATA
{
  vm:j39Pw744g5U rdf:type 'https://github.com/blacktiago/HSE/ontology/hse.owl#video' .
  vm:j39Pw744g5U vm:url 'http://www.youtube.com/watch?v=j39Pw744g5U' .
  vm:j39Pw744g5U vm:has_a _:j39Pw744g5U1 .
  _:j39Pw744g5U1 vm:show_a _:j39Pw744g5Ulizzielovesyoooh .
  _:j39Pw744g5Ulizzielovesyoooh vm:shows vm:anxiety.
  vm:anxiety rdfs:label 'anxiety'.
  _:j39Pw744g5Ulizzielovesyoooh foaf:name 'lizzielovesyoooh' .
  _:j39Pw744g5Ulizzielovesyoooh rdf:type 'http://xmlns.com/foaf/spec/#Person' .
  _:j39Pw744g5U1 rdfs:label 'escene_1' .
  _:j39Pw744g5U1 rdf:type 'https://github.com/blacktiago/HSE/ontology/hse.owl#frame' .
  _:j39Pw744g5U1 vm:frame_at '0:03' .
};
```

**Ilustración 22.** Ejemplo de inserción mediante Jena.

**Fuente:** Propia.

Fue necesario crear una consulta que permitiera extraer todas las instancias de videos de YouTube, que estuvieran relacionadas a determinada emoción; el proceso de ingresar una instancia en la ontología, no garantiza que el conocimiento sea válido; para poder acercar más el recurso a una base de conocimiento y aportar información relevante a la capa axiomática de una arquitectura de José Castillo et al. (2011), la información contenida dentro



del recurso ontológico tiene que ser verificable; para cumplir con el proceso de verificación de la información, resulta indispensable poder visualizar las instancias en función de determinada emoción.

```
PREFIX vm: <https://github.com/blacktiago/HSE/ontology/hse.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/spec/#>
SELECT ?videos ?url ?scene ?person ?label_emotion ?time ?emotion
WHERE {
?videos vm:has_a ?s .
?videos vm:url ?url .
?s rdfs:label ?scene .
?s vm:show_a ?p .
?s vm:frame_at ?time .
?p foaf:name ?person .
?p vm:shows ?emotion .
?emotion rdfs:label ?label_emotion .
?emotion rdfs:label 'anxiety' .
};
```

**Ilustración 23.** Ejemplo de consulta por emoción.

**Fuente:** Propia.

En el proceso de verificación de emociones de estrés humano en videos de YouTube, en determinado momento un operador requerirá eliminar un registro de la ontología, por considerarlo irrelevante o errado; se creó una sentencia SPARQL para dar soporte a la operación de borrado de registros en la ontología; en el caso de actualización de registros en recursos ontológicos; se la realiza mediante la combinación de las sentencias de borrado e inserción según lo indica la especificación del lenguaje SPARQL.

```
PREFIX vm: <https://github.com/blacktiago/HSE/ontology/hse.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/spec/#>
DELETE { ?person ?x ?y }
WHERE
{
?frame rdfs:label 'escene_1' .
?frame vm:frame_at '0:03' .
?frame vm:show_a ?person .
}
```

```

?person foaf:name 'lizzielovesyoooh' .
?person ?x ?y .
};
PREFIX vm: <https://github.com/blacktiago/HSE/ontology/hse.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/spec/#>
DELETE { ?frame ?prop ?y }
WHERE
{
vm:j39Pw744g5U ?prop ?frame .
?frame rdfs:label 'escena_1' .
?frame vm:frame_at '0:03' .
?frame ?prop ?y .
};

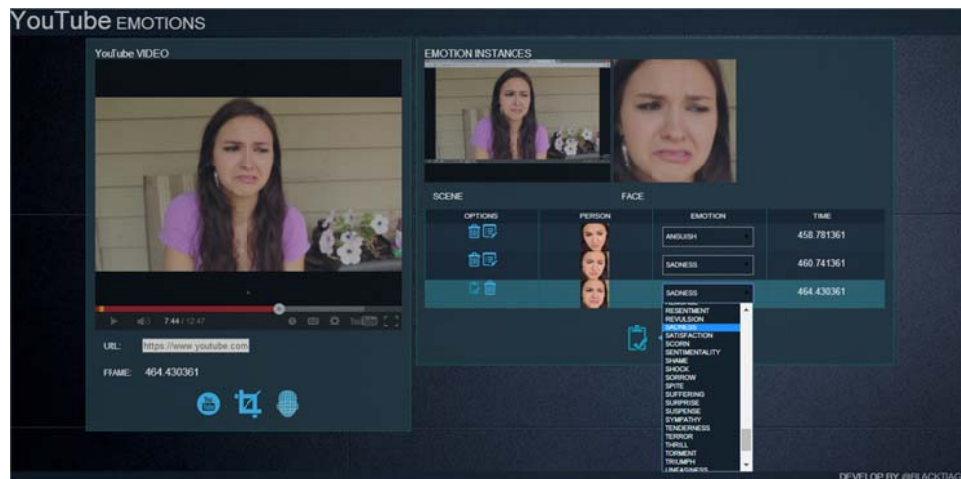
```

**Ilustración 24.** Ejemplo de borrado de entidades.  
Fuente: Propia.

## 4. Resultado del Aplicativo

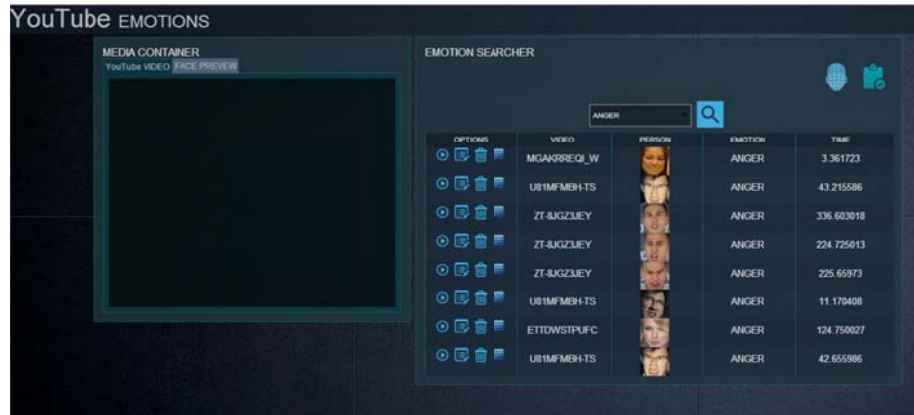
### 4.1. Mantenimiento de Instancias de Emociones en Videos

Para que el usuario pueda dar mantenimiento a las instancias de emociones en videos de YouTube se desarrolló una vista que muestra un control para visualizar un video e ir registrando las emociones, en la misma vista es posible eliminar y editar las instancias; se puede apreciar la vista en la Ilustración 25. Ingreso de emociones por video.



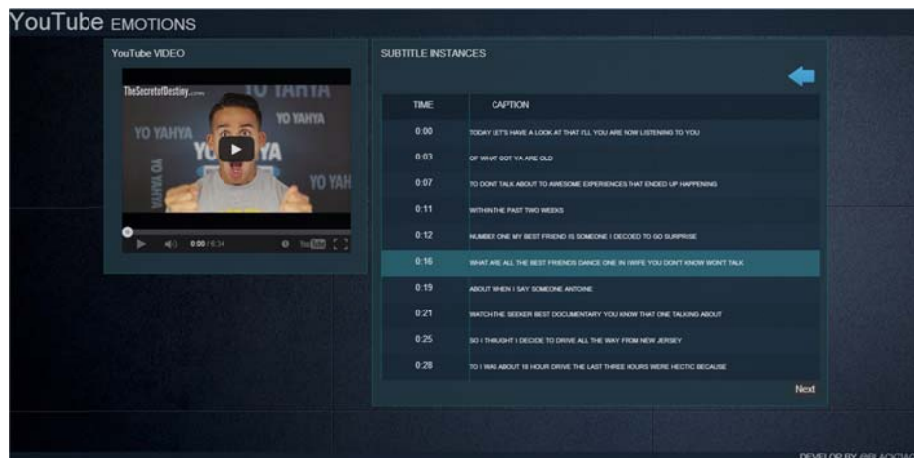
**Ilustración 25.** Ingreso de emociones por video.  
Fuente: Propia.

Como página principal del aplicativo se tiene una pantalla de consulta de videos por emoción desde la cual se puede guardar una nueva instancia o editar las ya existentes como se muestra en la Ilustración 26. Búsqueda por emoción.



**Ilustración 26.** Búsqueda por emoción.  
Fuente: Propia.

Para la poder visualizar el resultado de la extracción de subtítulos en el aplicativo se creó una vista que muestra el resultado, tal como se muestra en la Ilustración 27. Vista de extracción de subtítulos.

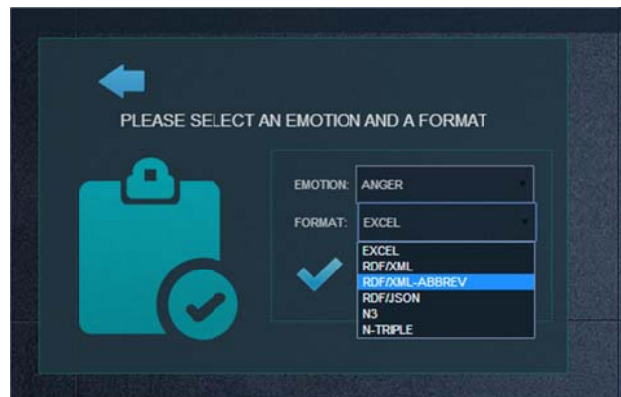


**Ilustración 27.** Vista de extracción de subtítulos.  
Fuente: Propia

#### 4.2. Exportación de Datos

La ontología de emociones humanas en videos de YouTube, proporciona un acercamiento a una base de conocimiento, que puede ser usada para el entrenamiento de redes neuronales y agentes inteligentes; la capa axiomática desarrollada mediante la metodología de José Castillo et al. (2011) será incorporada posteriormente a una red ontológica, que servirá para analizar información de diversos sensores de seguridad, para poder realizar inferencias que

permitan una respuesta oportuna en sistemas de vigilancia automática; la información contenida en la ontología, deberá ser procesada para encontrar patrones de interés, que relacionen una emoción con lo que la persona está diciendo en determinado momento; herramientas de análisis de datos como WEKA (University of Waikato, 2014), pueden ser usados en la extracción de patrones de interés, por lo que es necesario poder exportar los datos de emociones humanas a diferentes formatos. Para dar soporte a la necesidad de generar recursos, para la búsqueda de patrones de interés mediante técnicas de minería de datos y análisis de sentimientos, se incorporó en el aplicativo una vista XHTML que permite exportar datos por emoción a diferentes formatos como son: RDF/XML, RDF/XML-ABBREV, RDF/JSON, N3, N-TRIPLE, EXCEL; el soporte para los diferentes formatos fue implementado gracias a las librerías incorporadas en Jena (2014) a excepción del formato Excel, para lo que fue necesario completar la herramienta, mediante una clase que utiliza las librerías propias de Java para poder crear documentos de Excel.



**Ilustración 28.** Vista para exportado de datos.  
**Fuente:** Propia.

## 5. Trabajos Futuros

A futuro se espera incorporar la ontología con instancias de emociones para ser utilizado en aplicaciones de reconocimiento automático en imágenes y videos de diversas fuentes para ser utilizados en sistemas de vigilancia basados en reconocimiento facial; se espera también incorporar un módulo para el análisis de sentimientos sobre el texto obtenido de los subtítulos para poder relacionar expresiones faciales con expresiones típicas en diversas situaciones. El conocimiento almacenado y mantenido en la ontología requiere ser verificado para poder considerar el recurso ontológico como una base de conocimiento, para lograr este objetivo se espera poder implementar un método de verificación colaborativo. A futuro se espera poder llevar el aplicativo a plataformas móviles tanto como servicio de reconocimiento de emociones así como también para la verificación del conocimiento.

## CONCLUSIONES

- Se implementó satisfactoriamente un método inteligente mediante ontologías para gestionar la información emocional en videos de YouTube.
- Se evidencio la capacidad de una lógica descriptiva para dar soporte a operaciones transaccionales que permitan dar mantenimiento a un modelo ontológico mediante SPARQL.
- Una plataforma web se comporta adecuadamente como interfaz para procesos transaccionales sobre ontologías.
- Se logró elaborar un modelo ontológico que agrupa emociones humanas en tres niveles con múltiples jerarquías, que puede ser usado como referencia en la identificación automática de emociones.
- El uso de recursos ontológicos permiten el modelado de dominios de una manera más consistente con el orden real de las cosas.
- El uso de ontologías modulares en el proceso de diseño arquitectónico puede ayudar a identificar defectos de diseño, requerimientos faltantes o innecesarios en etapas tempranas del desarrollo, disminuyendo el costo.
- El uso de ontologías en el proceso de desarrollo contribuye a la creación de un modelo más consistente; un fichero que contiene un modelo y sus instancias en un formato ontológico posee información mucho más relevante que un archivo con sentencias SQL.
- Una misma entidad puede tener una apreciación distinta en función de cada dominio en el que se desenvuelve; el criterio de separación de un dominio debería considerar el lenguaje formal de la implementación de una entidad en un dominio ya que si dos dominios manejan la misma descripción formal de una entidad se incurre en la creación innecesaria de dominios.
- El proceso de creación de recursos ontológicos no puede ser realizado de manera artesanal y requiere de metodologías robustas que no necesariamente deben ser iguales a las metodologías de desarrollo tradicionales.
- La reutilización de recursos ontológicos permiten agilizar el proceso de modelado ontológico y permiten considerar puntos de vista faltantes y posibles errores de diseño en el proceso de modelado.
- YouTube constituye una fuente de datos muy completa de rostros humanos y conversaciones que pueden ser usados para el entrenamiento de redes neuronales y agentes inteligentes.

- Reconocer una amenaza de seguridad no es suficiente para que un sistema pueda emitir alertas de manera oportuna, es necesario buscar mecanismos que permitan a los sistemas anticiparse ante incidentes de seguridad.
- Los eventos registrados por sistemas de seguridad no siempre concuerdan; como resultado de ello los sistemas pueden enviar alertas equivocadas y no detectar situaciones de potencial interés.
- La utilización de algoritmos de proporción para el reconocimiento de emociones en rostros humanos no toma en consideración objetos y contextos de una imagen que pueden cambiar notablemente la interpretación de la emoción.
- No se puede estructurar todos los contextos en los que debe operar un sistema de seguridad, para poder responder de manera más inteligente ante las incidencias de seguridad se requiere de mecanismos de aprendizaje automático que permitan a los sistemas mejorar su capacidad de reconocimiento.
- Es necesario generar aplicaciones capaces de interactuar con humanos en entornos inteligentes: considerando la interacción con toda clase de interfaces o sensores existentes además de sistemas ya construidos; la comunicación entre sistemas puede ser crítica en el proceso de adquisición de datos o de envío de alertas.
- Las herramientas metodológicas y patrones de diseño en muchas disciplinas integran características métricas y cualitativas describiendo entidades sin ningún significado semántico, por tal motivo el proceso de diseño no aprovecha el conocimiento y experiencia de sus diseñadores.
- La modularidad en el diseño de ontologías permite mantener conjuntos de entidades disjuntos, lo que facilita el mantenimiento de la ontología; la aplicación de procesos ontológicos como extensiones definiciones afectan solo un dominio, manteniendo independiente los demás dominios y reduciendo el coste de un requerimiento de cambio.
- La utilización de una matriz emocional jerárquica permite el reconocimiento oportuno de emociones que pueden resultar en potenciales situaciones de riesgo para los sistemas de seguridad.

## RECOMENDACIONES

Se recomienda:

- Extender las fuentes de emociones a videos e imágenes de redes sociales, por ser contenido que posee un contexto más especializado en relación a una persona.
- Implementar un sistema de verificación de la información contenida en la ontología, para poder convertir el recurso en una base de conocimiento confiable.
- El uso de bases de datos gráficas para mejorar los tiempos de respuesta del aplicativo y poder implementar arquitecturas distribuidas.
- Implementar servicios para dar mantenimiento a las entidades de la ontología de emociones humanas en videos de YouTube, que permitan extender las operaciones a terminales móviles.
- La relación entre emociones observadas y objetos en una escena, pueden relacionarse de mejor manera reutilizando la ontología VERL.
- Unificar el registro de emociones mediante algoritmos de proporción facial al reconocimiento de emociones por contexto, para mejorar la precisión del reconocimiento.
- Preparar equipos de desarrollo especializados en herramientas como Jena y Maven para generar un mayor número de aplicaciones basadas en recursos ontológicos.
- Introducir módulos ontológicos en los desarrollos tradicionales para automatizar los procesos de recomendación y auto aprendizaje.

## **BIBLIOGRAFIA**



- Garlik. (2009). Obtenido de 4store.org
- Allen, J., & Ferguson, G. (1994). Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation*.
- Bindu, M. (2007). Cognitive Model - Based Emotion Recognition From Facial Expressions For Live Human Computer Interaction. *Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. IEEE Symposium on* , 351 - 356 .
- Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 2-4.
- Bremond, F., Maillot, N., Thonnat, M., & Vu, V.-T. (2004). *Ontologies For Video Events*. INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE.
- Carnegie Mellon University. (27 de 04 de 2014). *CMU Sphinx*. Obtenido de CMU Sphinx: <http://cmusphinx.sourceforge.net/>
- Castillo, J., Fernández, A., & López, M. (2011). A Framework for Multisensory Intelligent Monitoring and Interpretation of Behaviors through Information Fusion. *2011 Seventh International Conference on Intelligent Environments*, (págs. 1-4).
- DBpedia. (29 de 03 de 2014). *DBpedia*. Obtenido de DBpedia: <http://dbpedia.org/About>
- Fernández, A., Castillo, J., & Rodríguez, J. (2010). A Proposal for Local and Global Human Activities Identification. *6th International Conference, AMDO 2010, Port d'Andratx, Mallorca, Spain, July 7-9, 2010. Proceedings*, (págs. 78-87).
- Fonou-Dombeu, J. V., & Huisman, M. (2011). Semantic-Driven e-Government: Application of Uschold and King Ontology Building Methodology for Semantic Ontology Models Development. *International Journal of Web & Semantic Technology*.
- Gargoyle Software Inc. (2014). *Htmlunit*. Obtenido de <http://htmlunit.sourceforge.net/>
- Goguen, J., & Burstall, R. (Enero de 1992). Institutions: abstract model theory for specification and programming. *Journal of the ACM*, 39, 95-146.
- Gómez Pérez, A., Suárez Figueroa, M., & Fernández López, M. (10 de 03 de 2014). *NeOn Book*. Obtenido de [http://www.neon-project.org/nw/NeOn\\_Book](http://www.neon-project.org/nw/NeOn_Book)
- Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2009). NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology. *International Conference on Software, Services & Semantic technologies (S3T 2009)*. Sofia: Universidad de Sofia.
- Google. (04 de 27 de 2014). *Chrome Apps Office Hours: Text to Speech API*. Obtenido de <https://developers.google.com/live/shows/7320022-7001>
- Grobe, M. (2009). RDF, Jena, SparQL and the "Semantic Web" . *SIGUCCS '09 Proceedings of the 37th annual ACM SIGUCCS fall* .

- Grütter, R., Scharrenbach, T., & Bauer-Messmer, B. (2008). Improving an RCC-derived geospatial approximation by OWL axioms. (S. Berlin/Heidelberg, Ed.) *The Semantic Web-ISWC 2008*, 293-306.
- Handel, S. (24 de 05 de 2011). *The Emotion Machine*. Obtenido de <http://www.theemotionmachine.com/classification-of-emotions>
- Hoehndorf, R., Ngonga Ngomo, A.-C., & Herre, H. (2009). *Developing Consistent and Modular Software Models with Ontologies*. Leipzig.
- Hois, J., Bhatt, M., & Kutz, O. (2009). Modular Ontologies for Architectural Design. *Formal Ontologies Meet Industry*, 66-77.
- Kruchten, P. (2004). *The Rational Unified Process: an introduction*. Pearson Education.
- Kutza, O., Lutz, C., Wolter, F., & Zakharyashev, M. (Junio de 2004). Connections of Abstract Description Systems. *Artificial Intelligence*, 156.
- NeOn Proyect. (10 de 03 de 2014). *NeOn Book*. Obtenido de [http://www.neon-project.org/nw/NeOn\\_Book](http://www.neon-project.org/nw/NeOn_Book)
- NeOn Proyect. (10 de 03 de 2014). *Welcome to the NeOn Project*. Obtenido de <http://www.neon-project.org/>
- OMG. (2014). *Object Mangement Group*. Obtenido de <http://www.uml.org/>
- OpenLink. (2014). *Virtuoso*. Obtenido de <http://virtuoso.openlinksw.com/>
- Oracle. (27 de 04 de 2014). *JMF 2.1.1 Software Documentation*. Obtenido de <http://www.oracle.com/technetwork/java/javase/documentation-138769.html>
- Oracle. (2014). *The Java EE 6 Tutorial*. Obtenido de <http://docs.oracle.com/javaee/6/tutorial/doc/bnaph.html>
- Parker, M. M. (2010). Emotion and Stress. En M. M. Parker, *Understanding Psychology* (págs. 336-338). BVT Publishing.
- Parrott, W. G., & Schulkinb, J. (1991). *Neuropsychology and the cognitive nature of the emotions*. Lawrence Erlbaum Associates.
- Potencier, F. (20 de 04 de 2014). *La arquitectura MVC*. Obtenido de La arquitectura MVC: [http://librosweb.es/jobeeet\\_1\\_4/capitulo\\_4/la\\_arquitectura\\_mvc.html](http://librosweb.es/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html)
- Reenskaug, T. (2003). The Model-View-Controller (MVC) Its Past and Present. *JavaZONE*.
- Scrum. (2014). *Scrum.org*. Obtenido de <https://www.scrum.org/>
- Skuce, D. (1995). Conventions for reaching agreement on shared ontologies. *9th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff Conference*.
- Stanford University. (29 de 03 de 2014). *Protégé*. Obtenido de <http://protege.stanford.edu/>

- Sun Microsystems. (2008). *Speech Recognition: javax.speech.recognition* . Obtenido de <http://www.ling.helsinki.fi/kit/2004s/ct1310gen/L7-Speech/JSAPI/Recognition.html>
- The Apache Software Foundation. (2014). *Apache Jena*. Obtenido de <https://jena.apache.org/>
- University of Waikato. (29 de 04 de 2014). *Weka*. Obtenido de <http://www.cs.waikato.ac.nz/ml/weka/>
- Uschold, M., & King, M. (1995). *Towards a Methodology for Building Ontologies*. Edinburgh.
- W3C. (2008 de 06 de 15). *SPARQL Update*. Obtenido de SPARQL Update: <http://www.w3.org/Submission/SPARQL-Update/>
- W3C. (06 de 05 de 2000). *JSpeech Grammar Format*. Obtenido de <http://www.w3.org/TR/2000/NOTE-jsgf-20000605/>
- W3C. (28 de 10 de 2004). *XML Schema*. Obtenido de <http://www.w3.org/XML/Schema.html>
- W3C. (27 de September de 2006). *Time Ontology in OWL* . Obtenido de <http://www.w3.org/TR/owl-time/>
- W3C. (13 de 10 de 2011). *Ontology for Media Resources 1.0*. Obtenido de <http://dev.w3.org/2008/video/mediaann/mediaont-1.0/mediaont-1.0.html>
- W3C. (16 de 04 de 2014). *FOAF Vocabulary Specification 0.99*. Obtenido de FOAF Vocabulary Specification 0.99: <http://xmlns.com/foaf/spec/>
- W3C. (12 de 01 de 2014). *FOAF Vocabulary Specification 0.99*. Obtenido de FOAF Vocabulary Specification 0.99
- Wang, P., Lu, J., & Li, Y. (2004). Theory and semantic refinement of bridge ontology based on multi-ontologies. *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference*, (págs. 442 - 449).
- Youtube. (2014). *Google Developers*. Obtenido de <https://developers.google.com/youtube/?hl=es>

## **ANEXOS**

## 1. Anexo A

# Ontologías Modulares para el Diseño Arquitectónico

(Informe descriptivo)

### Introducción

Un proceso de diseño arquitectónico puede agrupar diversos niveles de detalle, existen herramientas en continuo desarrollo para solventar la necesidad de generación de diagramas, modelos, planos, maquetas y demás con el único propósito de reflejar un producto terminado; la apreciación de los diferentes componentes y la manera en que interactúan, permiten a sus diseñadores tomar decisiones en etapas tempranas, reconocer escenarios potencialmente riesgosos, estimar costos, recursos, calidad, tiempo, etc. Los elementos que componen un modelo son entidades muy básicas que asociadas a una metodología constituyen un plan para la realización de un proyecto y/o producto; sin embargo esta manera de resolver el diseño no permite incluir información semántica asociada a los principales actores de un modelo ni aprovechar la experiencia con la que cuenta el diseñador del modelo; (Hois, Bhatt, & Kutz, 2009) proponen dotar a las herramientas de información para ambientes arquitectónicos con la capacidad de manejar diversas vistas o perspectivas del dominio y específicamente asociarles un información semántica; formalizando las diferentes capas de una arquitectura se puede aprovechar la modularidad no solo en un contexto de módulos y componentes sino también a nivel de modularidad semántica.

E-conexiones es una teoría propuesta por (Kutza, Lutz, Wolter, & Zakharyashev, 2004) que básicamente trata de resolver el problema del grado de formalismos que debe tener el modelado de un sistema de inteligencia artificial ya que el comportamiento de un modelo híbrido es mucho más caótico que el de cualquiera de sus partes; en pocas palabras pretende responder que cantidad deseable de formalismos es realmente deseable. (Hois, Bhatt, & Kutz, 2009) usaron como base la teoría E-conexiones para proponer un diseño modular de las vistas de una arquitectura ya que las ontologías pueden a más de estar separadas de manera lógica también se encuentran separadas por dominios o tópicos.

### Ontologías modulares

En el amplio campo de la ingeniería ontológica, la modularidad se ha convertido en un aspecto indispensable y de este aspecto se desprenden tres preguntas elementales que caracterizan la modularidad ontológica:

- ¿Cómo se puede crear ontologías largas y complejas a partir de elementos ontológicos más pequeños; teniendo en cuenta que pueden estar en diferentes lenguajes lógicos? (Combinación)

El planteamiento anterior implica también que se requiere un proceso para relacionar cada parte y además genera la interrogante opuesta como se puede fraccionar una gran ontología (Popularización)

- ¿Cómo se puede representar una estructura ontológica modular sin perder las propiedades lógicas de cada una de sus partes?
- ¿Cómo y cuándo se puede realizar razonamiento sobre una estructura ontológica modular y cuando se debe hacer solo sobre alguna de sus partes?

Las preguntas anteriores apuntan a un tema común y es la representación; siendo el reto encontrar la mejor manera de representar cada módulo ontológico y poder reusarlo a futuro sin siquiera considerar aun los problemas que generen los módulos y el razonamiento que aparecerán con el diseño arquitectónico. La modularidad se viene utilizando por un largo tiempo existen toda clase de elementos y objetos que pueden ser integradas a un diseño arquitectónico terminado o también se puede considerar el escenario de ciertos elementos que pueden ser reemplazados sin repetir el proceso completo de construcción o el de diseño sin comprometer partes significativas del objeto; por ejemplo cuando se quiere mejorar la seguridad de un domicilio se colocan protecciones en puertas y ventanas o cuando un tejado ha cumplido su vida útil puede ser reemplazado; eso pensando en las formas más comunes de diseñar y construir edificaciones. Todos los cambios que se pueden hacer sobre estructuras arquitectónicas bajo las técnicas normales apuntan a cambiar su forma afectando en menor medida su propósito pero aún no se puede estructurar su comportamiento de manera modular y para poder lograr esta clase de componentes modulares son necesarias estructuras formales desde que describan a más del objeto la forma de construirlo y su comportamiento de tal forma que los productos terminados puedan crecer modularmente también a nivel de comportamiento; esto está estrechamente ligado con lo que se conoce como inteligencia ambiental.

### **Razonamiento formal y ontología modular**

Existen diversos procesos de ingeniería y herramientas desarrolladas que ya soportan el diseño de arquitecturas mediante la utilización de ontológicas; pero en la mayoría de los casos están orientadas a aspectos de muy funcionales y fueron pensadas para replicar procesos ya existente sin aprovechar de manera significativa las capacidades de una lógica descriptiva formal; en pocas palabras el mismo proceso en otro contexto; por tal motivo es importante redefinir el proceso de diseño para poder hacer uso de las bondades del razonamiento automático de lo contrario el nuevo proceso tendrá las mismas deficiencias.

La convergencia de la modularidad y los recursos ontológicos han generado algunas dimensiones que deben considerarse en el proceso de diseño y construcción de espacios

arquitectónicos y desde luego puede ser aplicado a muchos otros campos de diseño; esta forma de concebir el diseño supone nuevos retos para los procesos de razonamiento automático, algunas etapas del diseño propuesto así como los artefactos aún pueden ser fácilmente soportados por la mayoría de las herramientas existentes; a continuación se resumen algunas dimensiones de la modularidad ontológica y retos del razonamiento:

**Heterogeneidad Semántica:** También conocida como capa de lenguaje se refiere al problema que surge al combinar dos modelos ontológicos; aun si su están escritas bajo la misma lógica formal, no se tiene garantía de que la sintáctica y la semántica de ambos modelos coincida; en un contexto de personas, una personas en un modelo A puede estar definida de manera diferente en una ontología B, esto se conoce como reconciliación de vocabularios en ontologías; Para solucionar este problema se suele realizar un traducción de una lógica a otra complementando los valores faltantes durante el proceso de traducción esta práctica se conoce como teoría de la institución (Goguen & Burstall, 1992); que solo toma como referencia los modelos y sus relaciones ignorando los detalles de la naturaleza del sistema lógico que las maneja. En cuanto a la estructura que se genera se modelos preconcebidos o descritos para otros propósitos no relacionados con el significado de una entidad; un error común es pasar las definiciones exactamente iguales de un esquema de definición a otro; generando toda clase de problemas que en algunos casos incluso llevan a los razonadores a inferir de manera equivocada. Cuando se parte de un modelo previo, los datos manejados por el sistema estarán estructurados de cierta manera y en la mayoría de los casos se requiere de esta información para la toma de decisiones; los procesos de automatización o automatización parcial para la instanciación de los elementos de manera que sean entendidos por las ontologías creadas supone un alto coste y en muchos casos puede generar modelos relaciones inconsistentes o no soportadas tanto para el esquema origen como para el destino; las soluciones generadas solo a nivel de aplicación no aportan ningún valor semántico al proceso.

**Independencia Lógica:** Este es uno de los conceptos más importantes dentro de la modularidad ontológica y básicamente dice que una ontología A es una extensión conservadora de otra ontología B si todas las afirmaciones hechas en el idioma de la ontología A; la ontología B las conserva; se conoce como noción conservativa puede usarse para extraer módulos de una ontología más grande que sean lógicamente independientes. Es muy poco probable verificar la noción conservativa en lógicas de primer orden y expresiones de descripción lógica (DL); el caso más simple de extensión conservativa en una extensión definicional donde una entidad en la ontología A se le agregan nuevas definiciones pero sigue sujeta a los axiomas de A. El éxito o fracaso de esta dimensión modular ontológica está estrechamente ligada al criterio de integración del diseñador; en términos ontológicos se pueden hacer uso de diversos vocabularios y crear axiomas que

operen las entidades base, las entidades personalizadas o las combinaciones de entidades; el uso excesivo de vocabularios para una definición escasa de entidades compromete la independencia lógica del modelo; mientras que si no se utilizan vocabularios comunes las ontologías difícilmente podrán ser manipuladas en sus otras dimensiones por la inconsistencia del modelo.

**Coincidencia y alineación:** Al superponer dos ontologías, surge el problema de la identificación de la temática y de esto se ocupa el principio de coincidencia, cuando se relacionan ontologías mediante elementos comunes o sinónimos se está haciendo uso del principio de alineación; para el manejo de este tipo de relaciones se emplean métodos y heurísticas para encontrar una medida de cercanía o similitud entre las ontologías lo que supone un esfuerzo mayor que si se trataran con principios estructurales o de conservación donde la relación entre ontologías puede ser tratada mediante algunos postulados o axiomas. Cuando las ontologías resultantes de un proceso de diseño resultan tener un bajo grado de acoplamiento y satisfacen las restricciones de su modelado surge la necesidad poner a prueba la consistencia del modelo consolidado; los criterios usados para el cálculo de la similitud de sus instancias van más allá de una simple proyección; el bajo grado de acople de las entidades del modelo debería suponer suficiente dispersión como para realizar un cálculo de similitud apropiada, pero si se involucran atributos comunes en la mayoría de los casos no sucede así por lo que una categorización inadecuada puede resultar en la toma de una decisión sobre una instancia incorrecta.

**Integración y conexión:** Cuando dos recursos ontológicos A y B son integrados en un tercer recurso C cualquier operación existente en A o en B debe volver a ser interpretada por la ontología C; Este proceso no se encuentra formalizado del todo por la diversidad de formas en que se puede definir una entidad dentro de una ontología y a la correspondencia que tenga con otro recurso; sumado a esto la complejidad de la integración debe considerar que los dominios y rangos de entidades equivalentes pueden no ser compatibles. La integración es una aproximación a lo que se conoce como Integración Semántica que condensa dos ontologías mediante técnicas de mapeo o traducción; el fin último de una integración satisfactoria es una ontología que mantenga la integridad de las ontologías que referencia permitiendo de este modo tener un entendimiento común con otras ontologías que manejan las mismas definiciones.

La intención de tener diversas relaciones a través de múltiples dominios resumen la versatilidad del marco de trabajo propuesto por (Hois, Bhatt, & Kutz, 2009); la variedad de expresiones que dotan a cada lenguaje de construcción la capacidad de describir un objeto puede ser usado para conectar sus relaciones. El nivel de acoplamiento entre las definiciones conceptuales de dos o más dominios estarán definidos por la forma en la que el objeto es percibido por su entorno y plasmada por su lenguaje de descripción; un mismo



objeto puede tener diversas concepciones a lo largo de un ciclo de vida es decir; requiere una definición propia para cada dominio, y el paso entre dominios requiere un mapeo o una traducción que debe ser mantenida al nivel más bajo posible ocultando los detalles de la implementación de su medio y enfocándose en los atributos relevantes al dominio.

### **La modularidad de Especificaciones de Diseño Arquitectónico**

Un proceso de construcción debe ser procedimental y sustentado por un determinado número de artefactos que permitan tener diferentes perspectivas del proceso de construcción; cada perspectiva involucra un dominio para un objeto en construcción y por lo tanto es necesario realizar un proceso previo al diseño que recopile la información que permita tener un criterio de segmentación de dominios.

Para la entender el alcance y visionamiento de un diseño arquitectónico la modularidad ontológica hace uso de algunas directrices manuales como:

**Modulo temático:** Dado un dominio D, un módulo temático A se ocupa de los aspectos particulares de desde el punto de vista que D tienen de un objeto; esto nos da la noción de que es necesario tener dominios disjuntos cuando se tiene dos o más módulos de temática diferentes para el dominio D; por ejemplo: Si tenemos un espacio conceptual para los materiales que componen los objetos y otro espacio para las relaciones cuantitativas de los objetos con su espacio topológico; el sentido común nos lleva a suponer que no se debe superponer ambos espacios conceptuales; en contraste si dos dominios diferentes de un diseño arquitectónico poseen módulos temáticos iguales; los espacios conceptuales deberían ser integrados; un módulo temático no hace referencia específica a la estructura de una entidad sino más bien al rol que cumple dentro de un dominio.

**Extensiones Definicionales:** La definición de una ontología puede extenderse agregando nuevos conceptos a la firma de definición; al agregar nuevos conceptos hay que tener en cuenta que se está extendiendo la ontología solo de manera definicional, es decir que no supone nuevas instancias ni la integración o conexión con nuevas ontologías. Un espacio definicional puede estar disponible en una ontología pero esto no quiere decir que se haya integrada entidades o superpuesto axiomas; una extensión definicional realizada mediante firma de definición podría ser tomada como aplicación de la teoría de puente; sin embargo la diferencia principal radica en el hecho de que el propósito de la extensión es vincular nodos de un espacio definicional a nivel de instancia no de modelo; así por ejemplo una extensión definicional mediante firma podría ser una referencia a otra ontología que tenga la definición del fabricante de una puerta; podemos saber detalles adicionales del fabricante y la puerta a través del enlace pero esto no compromete la definición que se tiene de la puerta en la ontología original; por el contrario al agregar en cambio sí dos puertas de ontologías diferentes deben ser entendidas de igual manera y estar sujetas a las mismas restricciones

se debe considerar un proceso de integración mediante traducción o aplicando el proceso descrito por la teoría de puente.

**Módulos de enlace temático:** Las consideraciones de este módulo vienen dadas por el juicio experto de un humano en este caso de un arquitecto, verificando la consistencia relacional entre los diferentes módulos de temática. El proceso de integración a lo largo de cada módulo se logra mediante E-relaciones que definan un vocabulario para introducir las relaciones y axiomas adecuados de cada módulo temático; como se debe conectar un módulo a otro y como no se debe conectar, un grado deseable de formalismo en la descripción relacional determina el grado de expresividad para la descripción de las relaciones. Dicho de otra forma el grado de entendimiento que tengamos de un dominio dependerá de la complejidad de las interacciones de sus elementos. Al momento no existen razonadores e-Conexiones dedicados; por este motivo (Hois, Bhatt, & Kutz, 2009) usaron para el escenario de su prueba la lógica descriptiva del lenguaje OWL para realizar la codificación de la semántica de e-conexiones disjuntamente de cada uno de los diferentes dominios. El enlace de temáticas introduce un nuevo concepto local que afecta las sus restricciones, dominios y rangos; este concepto se conoce como operadores de e-conexión; pero una vez más gracias a los lenguajes descriptivos existentes y universalmente aceptados se puede dar solución a las restricciones entre enlaces temáticos; de este modo además de tener una lógica expresiva para cada dominio se tiene un protocolo por así decirlo de restricciones para las interacciones entre módulos temáticos; con esto se logra que al aplicar un extensión definicional, un módulo de enlace temático puede ser usado para ajustar la forma en que el dominio afectado entienda el objeto y realice inferencias sobre el mismo.

### **Modularidad Ontológica en el Diseño Arquitectónico**

En su investigación (Hois, Bhatt, & Kutz, 2009) presentan algunas ontologías enfocadas a ilustrar el marco de trabajo para el diseño arquitectónico señalando algunos aspectos generales de la formalización de su desarrollo; posteriormente introducen especificaciones ontológicas modulares métricas y cuantitativas en diferentes perspectivas conceptuales y de esta manera introducen criterios de diseño formales y aspectos relevantes a la modularidad para concluir presentan una evaluación de los aspectos relacionados a la expresividad y el razonamiento. En este punto conviene resumir lo el proceso de diseño introducido por (Hois, Bhatt, & Kutz, 2009) así:

#### **Aspectos generales de la especificación**

Una ontología puede ser definida mediante cualquier tipo de lógica (Hois, Bhatt, & Kutz, 2009) recomienda el uso de lenguajes ontológicos web OWL dl; los lenguajes de descripción OWL pueden ser de dos tipos: Abox y Tbox; En cuanto al tipo Tbox se caracteriza por contener las clases y sus relaciones además de las restricciones rangos y dominios;

mientras que las ontologías OWL tipo Abox contienen las instancias específicas generadas a partir de un clase; se puede decir que es la instanciación de entidades de un tipo Tbox.

De manera concluyente la necesidad de mantener ontologías separadas para el proceso de diseño arquitectónico; responde de una manera más inteligente que una distribución de vistas generada a partir de un concepto funcional (requerimiento); la intención de un diseño arquitectónico así pues debería centrarse en el modelamiento de las entidades y sus relaciones antes que del comportamiento funcional de una entidad; dotando de nivel semántico a un modelo basado en ontologías modulares, el requerimiento funcional será solventado por un axioma en un dominio.

Las Capas propuestas por el marco de trabajo de (Hois, Bhatt, & Kutz, 2009) se describen de manera práctica con un ejemplo de la construcción de un edificio:

### **Capa Cuantitativa**

Se plantean un escenario de construcción de una vivienda basados en planos y especificaciones de un modelo de datos IFC; las clases más elementales pueden ser reflejadas en la primera capa de información de la arquitectura; los elementos para la construcción de viviendas pueden ser tomados de la especificación de una ontología de construcción de edificios donde todos elementos de las construcciones están provienen de la clase BuildingElement; a nivel de capa cuantitativa.

### **Capa Cualitativa**

En la ontología de construcción de edificios en esta capa ofrece relaciones basadas en regiones espaciales como se define en el cálculo de región de conexión RCC. Re utilizando la ontología RCC propuesta por (Grütter, Scharrenbach, & Bauer-Messmer, 2008) donde las ocho relaciones entre regiones están definidas en una jerarquía para aplicar a propiedades específicas de las entidades en la capa cualitativa, por ejemplo los tipos de proximidad que una puerta pueden tener respecto a otras entidades. En un espacio arquitectónico pueden existir todo tipo de puertas que se ajusten a la definición anterior pero vale la pena destacar que el cálculo de una región puede determinar la aplicación de determinado acción; por ejemplo todas las puertas que conecten una habitación con un corredor deben tener por defecto el estado cerrado mientras que todas las puertas que conecten una cocina con un corredor o cualquier otro espacio debería permanecer cerradas; de este modo el cálculo de regiones es brinda la capacidad de diferenciar la aplicación de un axioma en función de su dominio; el ambiente determina el objeto y su comportamiento, mientras que si se usan conceptos de diseño tradicionales lo que se haría es crear un identificador para el objeto y guardarlo en una lista de objetos que deben cumplir la condición.

### **Capa Conceptual**

En relación a las entidades de una arquitectura, la capa conceptual se ocupa de las características idiosincráticas es decir de todas aquellas propiedades y axiomas que

determinan una entidad sin considerar un contexto y ninguna otra propiedad incrustada; por ejemplo una capa conceptual puede ser apreciada en una subclase de puerta como puerta principal. Las entidades de la capa conceptual pueden relacionarse no solo con aspectos del plano de construcción; sino también con clases abstractas que pudieran o no haber originado la clase hija como funciones operaciones y costos; de esta capa se generara una ontología fundamental para ir especificando las relaciones de un dominio; muy similar a un modelo de datos en otras disciplinas como la ingeniería de software donde la gran mayoría de artefactos se generan a partir de un modelo conceptual.

Para el manejo de las entidades en esta capa (Hois, Bhatt, & Kutz, 2009) utilizaron una ontología llamada DOLCE-Lite que básicamente categoriza y describe entidades arquitectónicas que se ven enriquecidas y descritas a un nivel más especificado; así por ejemplo podemos decir que la entidad ventana es diferenciada por estar elaborada de un solo material de un determinado tipo.

### **Representación integrada**

Uniendo las definiciones cuantitativas (Datos métricos de la construcción), cualitativa (relaciones funcionales de las entidades) y las especializaciones necesarias a nivel de concepto. En el diagrama anterior se define un objeto representable como una instancia particular que posee una definición sobre los datos o características de construcción y otra definición sobre que indica los aspectos funcionales de las mismas; hay que destacar q esta calase debe ser única y disjunta es decir; existe solo una manera de representar un objeto pero esto sucede solo a nivel de representación del modelo; la entidad o la apreciación de la misma sigue teniendo independencia a través de las capas anteriores del proceso de diseño. Las tres capas integradas en la representación anterior están es un estado artificial, siendo los nodos de más alto nivel de sus respectivas ontologías; los espacios de nombres indican los diferentes orígenes de sus propiedades; como resultado de esta representación las clases de estas ontologías permanecen disjuntas que por ser modelos no instancian ninguna clase específica, además la representación integrada no define ninguna nueva clase ni subclase menos aun instancias específicas; en lugar de ello describe los enlaces que existen entre entidades de diferentes ontologías; por ejemplo una clase específica de la capa cuantitativa como Door, en la capa cualitativa puede estar vinculada a más de una clase hija de Door diferenciada por algún atributo funcional; es decir que una puerta de ingreso a una habitación puede tener la misma dimensión pero sirve de acceso a diferentes ambientes.

### **Requisitos específicos de la tarea**

Se puede apreciar en la capa de representación integrada que se requiere definiciones y restricciones adicionales para complementar los requisitos específicos de un proceso sujeto de automatización; estas restricciones no solo son generados por los requerimientos sino

también por las necesidades generadas por el proceso de diseño como tal y en base a la información arquitectónica generada; de este punto se ocupa la capa de requisitos específicos de la tarea; formula requerimientos que deben ser atendidos de manera concreta con plan de construcción; las tareas descritas en esta capa especifican un propósito que debe ser cumplido; por ejemplo restringir el nivel de privilegios de un usuario, el tipo de dispositivo que puede tener acceso a determinado servicio, el tiempo que el servicio puede estar activo y condiciones que deben ser cumplidas para la prestación de un servicio, tareas de monitoreo y vigilancia, etc. Las restricciones a este nivel son formalizadas en ontologías que re utilizan los enlaces generados por la capa de representación de la integración para moverse a través de los dominios e interactuar con las entidades de cada capa; una de las ventajas de esta forma de representación es que abstraen el comportamiento funcional a muy alto nivel y lo desacoplan de la definición real de una entidad; haciendo esto lo que en ingeniería de software se conoce como lógica de negocio termina siendo manejado de una manera muy dinámica ajustándose mejor a la forma en que los sistemas en constante cambio interactúan en el día a día de las personas por ejemplo una restricción a este nivel podría ser que todas las construcciones tengan una terminal con inteligencia de navegación que puede orientar a los visitantes en diversos aspectos; tal requerimiento en el diseño tradicional implica desarrollar nuevos sistemas con sus propias definiciones y restricciones, y que tal si el sistema existe pero no es visible por el sistema de inteligencia ambiental; en este caso se deberá exponer servicios y establecer contratos para poder cumplir con el requerimiento y el resultado final probablemente no pueda interactuar en una manera completamente satisfactoria con las entidades involucradas; mientras que realizar el cumplimiento del requerimiento a nivel de capa de requisitos específicos de tarea solo implica la introducción de nuevos axiomas.

### **Escenario de Inteligencia Ambiental (Aplicación)**

Para ilustrar su investigación (Hois, Bhatt, & Kutz, 2009) plantean un escenario donde se aplica el marco de trabajo antes descrito; para dotar una edificación con inteligencia ambiental; teniendo como ámbito de mercado el hogar, negocios de todo tamaño, entornos académicos, sitios turísticos, administración, salud, agricultura, etc. La idea es enriquecer todos los aspectos del diario vivir dotando los ambientes en que nos desenvolvemos de un sistema de decisión automática capaz de responder según las condiciones del ambiente y las entradas de sus interfaces.

## 2. Anexo B

### Ontología de eventos en video

(Informe descriptivo)

#### Introducción

La necesidad de ontologías en el dominio del análisis e interpretación automática de video surge el aumento de sistemas orientados a esta tarea alrededor del mundo; una ontología agrupa conceptos, relaciones y restricciones de un dominio bajo un determinado nivel de convenciones de diseño, siendo los recursos ontológicos la principal herramienta de los expertos en la creación de sistemas que respondan de manera autónoma ante la detección de eventos en fuentes de video. Una ontología permite hacer entendible el concepto de video en sistemas enfocados en la experiencia del usuario final; y le permite al experto describir modelos muy complejos en términos entendible por las computadoras, así como también abstraer modelos ya existentes para un dominio de interés; la ontología es crucial al momento de describir un video ya que nos permite tener una mejor comprensión de los eventos específicos que deben ser reconocidos en un video.

#### Meta Conceptos para la descripción de objetos físicos

El concepto physical-objects se refiere a todos los objetos del mundo real que pueden ser identificados a través de una cámara; class es una definición abstracta de una realidad física que corresponde a la naturaleza de un objeto físico; por lo general la clase viene dada por la forma del objeto, por ejemplo personas, casas, autos y cualquier otro objeto físico. Los atributos de un objeto físico se introducen como attribute de un objeto físico son todas las características propias del objeto físico por ejemplo color, estatura, género y demás características determinantes, la propiedad liveliness de un objeto físico está relacionado con la forma en que un objeto expresa su movilidad y autonomía de un objeto físico.

La propiedad current-liveliness de un objeto físico es la característica observable de un objeto en una escena en un determinado momento y puede tomar dos valores: mobile y contextual; de ahí que cualquier objeto cuya propiedad current-liveliness se encuentre como mobile se reconocerá como mobile-object mientras que si la propiedad es contextual el objeto se reconocerá como contextual-object.

Un objeto será tomado como mobile-object si en su estado inicial el objeto es identificado en movimiento; es decir en un instante cero del tiempo el objeto se detectó en movimiento; el tipo de movimiento que se relacione al objeto sirve para asociarlo con la clase propia del objeto; así pues en función de su movimiento podemos deducir si son personas, puertas, juguetes, etc.

El concepto agent puede ser percibido como un objeto móvil cuya propiedad liveliness puede tener diferentes estados, como por ejemplo un objeto remotely-movable, programmable, autonomous, etc. Por tal motivo un objeto tipo agent será considerado como un objeto tipo mobile-object.

Si un objeto es de tipo contextual-object, implica que no puede cambiar su posición en una escena a menos que un objeto tipo mobile-object este desplazándolo; un objeto tipo contextual-object en una escena puede estar sujeto a los siguientes estados: automatically-movable-at-the-same-position, fixed, static, movable-at-the-same-position, remotely-movable-at-the-same-position, displaceable.

Los estados antes mencionados por lo general se corresponderán con puertas, ventanas, muebles, arboles el fondo de la escena como tal; pero en el transcurso del tiempo su estado variante pondrá en duda la clase del objeto; así pues, una forma clara de discernir entre las clases de los objetos, es la percepción de su estado inicial en el proceso de adquisición. Si se tiene por ejemplo un carro sin conductor, determina al carro como un contextual object, mientras que si el carro tuviera conductor y estuviera en movimiento, la consecuencia es que el vehículo sea catalogado como un mobile-object; en este contexto la clase de un objeto no viene dado por forma sino por el estado de su propiedad liveliness, por lo tanto un objeto físico no puede cambiar su clase ni su liveliness durante la realización de una actividad, como se vio en el ejemplo del carro y el conductor, existe la posibilidad de que dos objetos se combinen para resultar en otro objeto que puede cambiar su liveliness a un grado más autónomo, siempre que el liveliness de un objeto pueda cambiar. Analizando otro caso como el de un carro de control remoto; podemos identificar claramente que se trata de un comportamiento tipo remotely-movable; sin embargo eso lo sabemos por qué de antemano se conoce el concepto del objeto; se debe considerar el carro a control remoto como un mobile-object debido a que el movimiento que se genera por acción de la persona que lo controla es un comportamiento observable en el objeto carro a control remoto; así pues cuando alguien juega con el carro le da la categoría de mobile-object y cuando esta estático es decir su propiedad liveliness es static es un contextual object.

La granularidad es otro punto a considerar cuando se habla de reconocimiento de objetos; si consideramos un grupo de personas cruzando la calle esto podría interpretarse como un mobile-object único en lugar de múltiples mobile-objects; se debe considerar además la complejidad de algunos objetos, una persona por ejemplo es capaz realizar simultáneamente muchas diferentes acciones con cada parte de su cuerpo, según la acción que realiza cada parte del cuerpo puede considerarse como un objeto móvil; por lo tanto se debe tomar en cuenta que un número finito de objetos tipo mobile-object pueden formar parte de otro objeto móvil pero estos objetos deben considerarse como un ente único.

Los objetos que se pueden encontrar en una escena pueden ser divididos mediante herencia; y cada clase de objetos puede tener su propia jerarquía de sub clases; a nivel general se puede tener objetos físicos como subclases de mobile-object: un individuo, un carro, un grupo de personas moviéndose, etc. Así mismo puede existir subclases de objetos físicos que pertenezcan a una subclase de objetos tipo contextual object como: objetos, portables, equipos, etc.

Role es un atributo de cualquier objeto físico que determina el comportamiento del objeto basado en determinadas circunstancias; le da un contexto por así decirlo; por ejemplo el comportamiento de una persona no será el mismo en un día lluvioso que un día soleado.

Existen muchos otros atributos observables en objetos físicos como velocidad, posición y otros que son clasificados como visual-attributes y pueden ser de tres tipos: position-based (basados en posición), global-appearance (apariencia global), local-appearance (apariencia local); visual-attributes de tipo position-based pueden ser trayectoria, posición, dirección; visual-attributes de tipo global-appearance pueden ser altura, color, tamaño; visual-attributes de tipo pueden ser: postura, rostro, etc.

### **Meta conceptos y descripción de actividades**

Existen diversas caracterizaciones de mobile-objects en una escena que dependen de la manera que los objetos interactúan y evolucionan como: evento, estado o actividad. En términos de herencia se conocen cuatro maneras de describir la relación que una clase puede tener como subclase:

- Especialización entre la clase y sus sub-clases.
- Composición de las subclases o comportamientos de los objetos físicos de una clase.
- Representación basada en relaciones espacio-temporal entre objetos físicos y sus atributos.
- Relaciones lógicas y temporales entre los estados y los eventos.

La propiedad state se refiere a un al conjunto de todas las posibles apreciaciones de un objeto en un intervalo de tiempo dado que caracteriza de manera unívoca a un objeto tipo mobile-object o a un mobile-object con respecto a otros objetos físico.

La propiedad primitive-state se refiere a estados espacio temporales dados en un intervalo de tiempo que resultan de la inferencia de los atributos visuales del objeto procesado por componente computacionales; la mayoría de aplicaciones orientadas al procesamiento de videos poseen las definiciones más básica de esta propiedad y suelen venir expresadas como atributos numéricos.

La propiedad composite-state se refiera a la unión de todos los estados primitivos entendidos como un sola unidad de procesamiento; en este punto existe un elevado grado de granularidad de los estados de un objeto; se conoce como componentes a todos los sub



estados de un composite-state y constraints a todas las relaciones que se pueden dar entre los objetos físicos, solo las relaciones de representación basada en relaciones espacio-temporal y relaciones lógicas y temporales entre los estados y los eventos figuran dentro de las constraints de un composite-state.

Se conoce como event al o los cambios de estado que pueden darse a lo largo de un intervalo de tiempo o entre dos instantes continuos de tiempo.

Un cambio de estado se conoce como primitive-event teniendo en cuenta que un evento es mucho más abstracto que un estado; hay que notar que la granularidad de un primitive-event está mejor definida y por lo tanto permite un mayor grado de diferenciación.

Un composite-event es una agrupa varios estados y eventos, lo que resulta en un sentido perjudicial para la granularidad de un evento; típicamente viene dado por valores simbólicos/booleanos vinculados a los requerimientos de una aplicación.

Un single-agent-event es una clase particular de evento en la que interactúa solo un objeto tipo mobile-object, en este punto se puede considerar a un grupo de personas que realizan el mismo evento como un evento de este tipo; esto es por la dificultad del reconocimiento granular de los objetos.

Un multi-agent event debe involucra al menos dos objetos del tipo mobile-object y cada objeto debe tener su propio rango de movimiento.

Un activity es un grupo de interrelacionados cuya combinación no está del todo definida; la composición de eventos es una actividad sujeta a objetos físicos con las mismas restricciones, este concepto permite la asociación de objetos como personas por ejemplo que están realizando la misma interacción en una escena; este nuevo concepto está relacionado a los anteriores pero define todo un rango de aplicaciones enfocados a situación y la severidad de las mismas.

Un objeto action es una clase particular de evento relacionada al cumplimiento de una tarea específica.

Un objeto situation es un estado que involucra a varios objetos físicos bajo determinadas circunstancias.

Un objeto process es una composición de eventos que los agrupa de una manera estrechamente acoplada.

Los conceptos chronicle, escenario y plan son eventos compuestos para el entendimiento de dimensiones tiempos; diferentes perspectivas espacio temporales, pueden ser testigos de un evento; por ejemplo un corredor puede ser visto como un evento de persona corriendo, como evento compuesta de secuencias de aceleración, velocidad y otros atributos relacionados a su movimiento; o puede ser visto como un evento multi-agente donde una pierna derecha es comparada con el movimiento de una izquierda. Finalmente la

apreciación de esta dimensión vendrá dado por la información de interés para el usuario del sistema.

### **Relación entre conceptos**

Las diferentes relaciones descritas plantean escenarios entre objetos de tipo mobile-Object y describen los aspectos de interés referentes a su interacción; pero aún no se ha introducido el propósito y la manera de interacción para objetos de tipo contextual-object. Las relaciones entre evento y estado así como las relaciones entre objetos físicos y sus atributos indica la manera en la que se deben observar las relaciones estado / evento y objeto atributo así pues se pueden encontrar relaciones basadas en observación, espacio, tiempo y espacio-temporal. Cuando se analizan las relaciones basadas en la observación se requieren filtros de espacio y tiempo además de operadores aritméticos y estadísticos. Las relaciones basadas en el espacio requieren consideraciones topológicas como distancia y geometría. Las relaciones espacio-temporales están dados por los cambios de las relaciones espaciales a través del tiempo. Las relaciones temporales también tienen eventos asociados a objetos físicos.

En términos de relaciones entre eventos se pueden catalogar de dos maneras lógica y temporal; las relaciones lógicas involucran operadores como “and” y “or”; para el tratamiento de relaciones temporales (Bremond, Maillot, Thonnat, & Vu, 2004) señala en su investigación que se requieren operadores del algebra de Allen y algunas relaciones cuantitativas referentes al comienzo y termino de un evento. Existen muchos tipos de relaciones entre eventos, siendo la más común una secuencia; también pueden existir iteración, interrupción, reanudación, etc. Se define una relación espacial como la interacción de objetos físicos que interactúan en un evento dado. Cuando se describe un evento compuesto, pueden considerarse sub-eventos como una opción útil, porque el carácter de opcional puede ser entendido como un coeficiente binario que indica si el sub evento es necesario o es opcional; esta capacidad puede ser muy útil a la hora de realizar descripciones de eventos compuestos.

### **Descripción de conceptos**

PhysicalObjects: Declara un concepto asociado a la realidad de un objeto físico; la clase que lo define y el rol asociado a la clase, solo deberían ser declarados aquellos objetos físicos que pueden cambiar a través de las diferentes ocurrencias de un concepto.

Components: Agrupa todos los componentes de un concepto y no es indispensable su declaración.

ForbiddenComponents: Declara un dominio de eventos que no debería suscitarse durante la ocurrencia de un concepto, también es opcional.

Constraints: Establece las condiciones entre los componentes y componentes prohibidos que definen un concepto.

### **Utilización de la ontología**

Existen dos maneras de usar la ontología propuesta; la primera es de manera descriptiva para poder obtener un modelo que refleje una realidad física; la segunda manera de utilizarla es guardando instancias de ocurrencias de una situación en video; resulta casi imposible describir todas las circunstancias que podrían darse en un determinado dominio; pensado granular-mente en cada circunstancia posible como por ejemplo: una persona haciendo preguntas, un niño llorando, un empleado lavando las ventanas, alguien levantado un bolígrafo del suelo; son todas circunstancias largas definiciones y vocabularios completos para describir los objetos de estos escenarios; pensar solamente en toda la gama de movimientos que se necesita describir de cada parte del cuerpo para abstraer el simple proceso de lavar la ventana, nos da una idea de la complejidad de la tarea.

Por razones prácticas se podría listar los estados y eventos primitivos de manera muy genérica, con un bajo nivel de granularidad en los estados que un objeto físico puede tener, además no es necesario describir de manera detallada todas las propiedades de un objeto, a nivel de implementación sin embargo se puede identificar un amplio rango de estados y eventos en videos mediante técnicas de cuadro delimitador y atributos basados en la posición; todo dependerá de los requerimientos de un sistema; la utilización de medios automáticos de reconocimiento tiene sus propios problemas; así por ejemplo el concepto de sostener un objeto puede tener un rango exageradamente amplio de posturas en las que una persona sostiene un objeto además de que la percepción de este objeto puede variar dependiendo de la posición del mismo. Para definir el modelado (Bremond, Maillot, Thonnat, & Vu, 2004) al final propone tres dimensiones a analizar; la riqueza que determina cuantos objetos y relaciones tiene una ontología; la profundidad que determina el grado de especialización que debe tener una jerarquía de clases; y la anchura que determina la cantidad de variaciones que deberían ser consideradas para una actividad.

### 3. Anexo C

## Time Ontology OWL

(Informe descriptivo)

### Introducción

La información relativa al tiempo está relacionada con todo tipo de aspecto de mundo real independiente-mente de la forma en que estas se manejen en la web; aunque existen sistemas y aplicaciones manejando este tipo de información; la forma en que se tratan los datos temporales dificultad mucho el proceso de extracción de conocimiento; adicional a ello existen diversos niveles de información relativa al tiempo que depende de muchos factores como la zona horaria, el formato de la fecha, la forma de referenciar un evento; por este motivo (W3C, 2006) Realiza la especificación de una ontología para el manejo de eventos y datos referentes al tiempo en el contexto de aplicaciones y servicios web; y que puede extrapolarse a situaciones del mundo real.

Si se considera por ejemplo que se tiene una video conferencia con personas en otro uso horario; según el sitio y el punto de referencia un asistente deberá adelantarse o atrasarse y por cuanto tiempo en su reloj para poder asistir a la conferencia; de ahí la necesidad de crear elementos ontológicos que puedan ser usados para el razonamiento en esta clase de situaciones. Otro tipo de situaciones que requieren el manejo inteligente del tiempo pueden ser por ejemplo; el análisis de tiempos por referencia; un evento comúnmente se lo puede expresar con expresiones tipo; “se graduó n años después de que ingreso a la carrera”; la expresión propuesta no relaciona dos fechas como tales o compara los elementos día ms y año de una fecha; si no que establece un punto de referencia y una condición para determinar la fecha; el termino condición en este contexto nos hace pensar en acciones de una ontología.

### Sintaxis

Para ilustrar las definiciones introducidas por la ontología; se utiliza la notación N3 también conocida como no-XML; es una forma más abstraída y de fácil entendimiento que describe a una ontología.

### Relaciones temporales topológicas

Como clase principal se principal de la ontología se tiene la clase conocida como entidad temporal, esta entidad solo tiene dos subclases: Instante e Intervalo; la diferencia principal entre los dos conceptos introducidos puede resultar fácil de reconocer; Instante hace

referencia a un momento del tiempo sin longitud; es decir que su punto de inicio y fin son el mismo; mientras que el objeto Intervalo se define como dos puntos de tiempo separados por un intervalo de tiempo.

Las entidades relacionadas con el tiempo pueden tener un principio y un final y para identificar estas relaciones únicas de una entidad temporal se usan las propiedades `hasBeginning` y `hasEnd` respectivamente; cuando el valor de la propiedad `hasBeginning` tiende al infinito positivo; se dice que una entidad no tiene fin; mientras que cuando su aproximación sucede hacia el infinito negativo, se dice que el objeto no tiene comienzo.

La relación `inside` describe la relación entre un instante y un intervalo; la creación de esta entidad no apunta a concebir un intervalo como una secuencia finita de instantes; sino más bien como una forma de describir la precedencia; que también hace referencia a la propiedad `before` así por ejemplo podemos decir que si una entidad temporal A sucede antes de una entidad temporal B; el final de A sucede antes del comienzo de B; se puede tomar un instante como una derivada de intervalo.

Las relaciones temporales pueden expresarse de una manera muy sencilla en términos de identidad; existe un cálculo orientado a las relaciones binarias entre espacio y tiempo propuestas por (Allen & Ferguson, 1994) que define a los intervalos propios como aquellos intervalos principio y fin son diferentes.

## **Duración**

Pueden existir muchas maneras de definir un intervalo; por ejemplo un día 9 horas 5 meses 20 años y muchas más; tradicionalmente el tiempo de información asociada al tiempo de una entidad se guarda en función de la información relativa a la entidad; por ejemplo la edad de una persona se da en años pero su fecha de nacimiento incluye un día un mes y un año determinadas; adicional a ello existen diversas formas de dar formato a los datos relacionados al tiempo. Resulta mucho más útil expresar información temporal de manera independiente del objeto asociado, para la ontología de tiempo se define un arreglo con 8 predicados para el elemento `DurationDescription`.

Un objeto temporal compuesto es aquel que se da en más de una escala como por ejemplo: 2 días 6 horas, 1 año 3 meses, 12 horas 20 minutos, etc. La entidad `DurationDescription` debe expresar cualquier objeto temporal complejo en una sola unidad.

La ontología del tiempo posee dos conjuntos de propiedades `DurationDescription` y `DateTimeDuration`; La diferencia principal entre las dos propiedades es el rango; así por ejemplo el rango para `DurationDescription` puede venir en decimales (6 años) mientras que para la propiedad `DateTimeDuration` puede venir como un tipo `xsd:gYear`. La forma de representación de la duración de los objetos temporales puede ser mapeada fácilmente a otras aplicaciones servicios y formas de representación de tiempo.

Para vincular una entidad temporal `TemporalEntity` Se utiliza la propiedad `hasDurationDescription`.

Un objeto tipo `DurationDescription` requiere algunas restricciones básicas como por ejemplo en lo referente a la cardinalidad podemos decir que la propiedad `years` solo puede tener un valor; cuando damos referencia temporal de un suceso; este suceso no puede tener dos años o dos meses o dos días; en términos de cuando paso mas no de cuanto duro; a continuación se muestran algunas restricciones sobre como limitar la cordialidad de las propiedades del objeto `DurationDescription`.

Las restricciones sobre los rangos de las propiedades pueden ser usados también como indicador de granularidad; así por ejemplo si se especifica que los días tengan cardinalidad 0; el nivel de granularidades deberá especificarse en años por ejemplo ya que el objeto `DurationDescription` no podrá especificar días. En este punto se debe tener en cuenta q las restricciones de cardinalidad sobre las propiedades hacen referencia a la duración del tiempo y no al formato de una fecha.

### **Zonas Horarias**

La zona horaria implica el que la hora del día que un observador tiene en un determinado punto del globo; no es la misma que para otro; algunos de los componentes de un objeto temporal son sensible a la zona horaria ; las horas son relativas, los minutos también desde que Australia central no está alineada al esquema horario UTC, los segundos no son relativos a la zona horaria; los días semanas meses y son valores que también resultan relativos; la implicación de la relatividad horaria en función de su espacio geográfico y su estandarización, implican la interacción con una ontología de geo localidades. Tradicionalmente lo que se hace en sistemas y servicios en línea es trabajar con estandarizaciones propias de cada zona horaria; la complejidad de manejar equivalencia temporales de zonas horarios no es de interés de la ontología propuesta por (W3C, 2006); que más bien trata de introducir una manera genérica de definir el tiempo independientemente de la localización; la tarea de llevar un esquema a otro es trabajo de personalizaciones ontológicas o del nivel de aplicación.

### **Descripción de fecha y hora**

Para el manejo de valores temporales relativos a la hora; es necesario primero definir una unidad de tiempo; esto se traduce en la ontología como un restricción que implica que la hora vendara dada en una unidad y solo una; la restricción referente a la unidad de la hora introduce el concepto `TemporalUnit`.

Así por ejemplo un tipo de unidad para una descripción temporal que haga referencia a las 11:00 am puede venir dada por minute; mientras que un tipo de unidad para la fecha 14 de

abril del 2013 puede venir expresada en términos de días como unidad; el tipo de unidad temporal debe ser una propiedad mandataria y todas las propiedades inferiores al tipo de unidad; si por ejemplo la una fecha viene con un tipo de unidad día; las propiedades hora minuto y segundo serán ignoradas; esto no supone en pérdida de datos por que debemos recordar que un objeto temporal puede estar asociado a un DurationDescription como también a un DateTimeDescription a través de las propiedades hasDurationDescription y hasDateTimeDescription.

La propiedad DateTimeInterval siendo una subclase de ProperInterval requiere una definición más completa de ahí que aunque cualquier objeto temporal tiene duración solo DateTimeInterval tiene DateTimeDescription; por ejemplo al referirnos a un fecha específica como 3 de marzo; se trata de un objeto temporal que posee DateTimeDescription mientras que al pensar en 3 marzo 12:30 – 4 marzo 12:30 hace referencia a un intervalo de un día y puede ser expresado mediante la propiedad inDateTime o inXSDDateTime.

Para los días de la semana existe la propiedad dayOfWeek y el dominio para esta propiedad es DayOfWeek.

La forma de especificación de DateTimeDescription puede servir por ejemplo para determinar restricciones en relación a constantes relacionadas objetos de este tipo; por ejemplo se puede definir que enero tiene como una constante ser el mes 1, es decir su propiedad unitType debe provenir de objetos tipo unitMonth y que su valor es 1. Para hacer referencia a un instante determinado de un intervalo de tiempo se tiene la propiedad inDateTime definida de una manera muy parecida a la propiedad hasDateTimeDescription.

La definición de inDateTime debe ser usado particularmente para situaciones que no hacen referencia a un instante específico del tiempo; por ejemplo si se dice que una clase empieza a las 3 de la tarde; se hace referencia a cualquier instante entre las 3:00 y las 3:01 de la tarde; por lo tanto no se refiere a un instante como tal; así la propiedad inDateTime es usada en estos casos donde no se hace referencia a un instante único sino más bien a un aproximación temporal a un instante. En OWL existen dos propiedades referentes al tiempo xsdDateTime y inXSDDateTime, cuya diferencia con las propiedades propuestas hasDateTimeDescription e inDateTime radica principalmente en el rango; para las primeras se usa un rango de datos definidos en un esquema simple de datos XML mientras que para el rango de las propiedades de OWL-Time son objetos de tipo DateTimeDescription.

#### 4. Anexo D

### Definiciones Ontológicas para emociones humanas

emotion	definition	parrott_level	es_label	parrotts_Child_Of
Love	<a href="http://dbpedia.org/resource/Love">http://dbpedia.org/resource/Love</a>	Primary	Amor	
Affection	<a href="http://dbpedia.org/resource/Affection">http://dbpedia.org/resource/Affection</a>	Secondary	Afecto	Love
Adoration	<a href="http://dbpedia.org/resource/Adoration">http://dbpedia.org/resource/Adoration</a>	Tertiary	Adoración	Affection
Fondness	<a href="http://dbpedia.org/resource/Fondness">http://dbpedia.org/resource/Fondness</a>	Tertiary	Afición	Affection
Liking	<a href="http://dbpedia.org/resource/Liking">http://dbpedia.org/resource/Liking</a>	Tertiary	Gusto	Affection
Attractiveness	<a href="http://dbpedia.org/resource/Attractiveness">http://dbpedia.org/resource/Attractiveness</a>	Tertiary	Atracción	Affection
Caring	<a href="http://dbpedia.org/resource/Caring">http://dbpedia.org/resource/Caring</a>	Tertiary	Cuidar	Affection
Tenderness	<a href="http://dbpedia.org/resource/Tenderness">http://dbpedia.org/resource/Tenderness</a>	Tertiary	Ternura	Affection
Compassion	<a href="http://dbpedia.org/resource/Compassion">http://dbpedia.org/resource/Compassion</a>	Tertiary	Compasión	Affection
Sentimentality	<a href="http://dbpedia.org/resource/Sentimentality">http://dbpedia.org/resource/Sentimentality</a>	Tertiary	Sentimentalismo	Affection
Lust	<a href="http://dbpedia.org/resource/Lust">http://dbpedia.org/resource/Lust</a>	Secondary	Lujuria	Love
Desire	<a href="http://dbpedia.org/resource/Desire">http://dbpedia.org/resource/Desire</a>	Tertiary	Deseo	Lust
Passion	<a href="http://dbpedia.org/resource/Passion">http://dbpedia.org/resource/Passion</a>	Tertiary	Pasión	Lust
Infatuation	<a href="http://dbpedia.org/resource/Infatuation">http://dbpedia.org/resource/Infatuation</a>	Tertiary	Amor Ciego	Lust
Longing	<a href="http://dbpedia.org/resource/Longing">http://dbpedia.org/resource/Longing</a>	Secondary	Anhelo	Lust
Longing	<a href="http://dbpedia.org/resource/Longing">http://dbpedia.org/resource/Longing</a>	Tertiary	Anhelo	Lust
Joy	<a href="http://dbpedia.org/resource/Happiness">http://dbpedia.org/resource/Happiness</a>	Primary	Alegría	
Cheerfulness	<a href="http://dbpedia.org/resource/Cheerfulness">http://dbpedia.org/resource/Cheerfulness</a>	Secondary	Alegría/Jovialidad	Joy
Amusement	<a href="http://dbpedia.org/resource/Amusement">http://dbpedia.org/resource/Amusement</a>	Tertiary	Regocijo	Cheerfulness
Bliss	-	Tertiary	Dicha	Cheerfulness
Gaiety	<a href="http://dbpedia.org/resource/Gaiety">http://dbpedia.org/resource/Gaiety</a>	Tertiary	Animación	Cheerfulness
Glee		Tertiary	Júbilo	Cheerfulness



Jolliness	<a href="http://dbpedia.org/resource/Jolliness">http://dbpedia.org/resource/Jolliness</a>	Tertiary	Felicidad/Jovialidad	Cheerfulness
Joy	<a href="http://dbpedia.org/resource/Happiness">http://dbpedia.org/resource/Happiness</a>	Tertiary	Alegría	Cheerfulness
Joviality		Tertiary	Jovialidad	Cheerfulness
Delight		Tertiary	Deleite	Cheerfulness
Enjoyment	<a href="http://dbpedia.org/resource/Enjoyment">http://dbpedia.org/resource/Enjoyment</a>	Tertiary	Disfrute	Cheerfulness
Gladness		Tertiary	Alegría/Regocijo	Cheerfulness
Happiness	<a href="http://dbpedia.org/resource/Happiness">http://dbpedia.org/resource/Happiness</a>	Tertiary	Felicidad/Dicha	Cheerfulness
Jubilation		Tertiary	Jubilación	Cheerfulness
Elation		Tertiary	Rlación/Júbilo	Cheerfulness
Satisfaction		Tertiary	Satisfacción	Cheerfulness
Ecstasy	<a href="http://dbpedia.org/resource/Ecstasy_(emotion)">http://dbpedia.org/resource/Ecstasy_(emotion)</a>	Tertiary	Éxtasis	Cheerfulness
Euphoria	<a href="http://dbpedia.org/resource/Euphoria">http://dbpedia.org/resource/Euphoria</a>	Tertiary	Euforia	Cheerfulness
Zest		Secondary	Ánimo	Joy
Enthusiasm	<a href="http://dbpedia.org/resource/Enthusiasm">http://dbpedia.org/resource/Enthusiasm</a>	Tertiary	Entusiasmo	Zest
Zeal		Tertiary	Celo	Zest
Excitement		Tertiary	Emoción	Zest
Thrill		Tertiary	Emoción/Conmoverse	Zest
Exhilaration	<a href="http://dbpedia.org/resource/Exhilaration">http://dbpedia.org/resource/Exhilaration</a>	Tertiary	Regocijo/Acento	Zest
Contentment	<a href="http://dbpedia.org/resource/Contentment">http://dbpedia.org/resource/Contentment</a>	Secondary	Contento	Joy
Pleasure	<a href="http://dbpedia.org/resource/Pleasure">http://dbpedia.org/resource/Pleasure</a>	Tertiary	Placer	Contentment
Pride	<a href="http://dbpedia.org/resource/Pride">http://dbpedia.org/resource/Pride</a>	Secondary	Orgullo	Contentment
Triumph		Tertiary	Triunfo	Contentment
Optimism	<a href="http://dbpedia.org/resource/Optimism">http://dbpedia.org/resource/Optimism</a>	Secondary	Optimismo	Contentment
Eagerness		Tertiary	Ansiedad	Contentment
Hope	<a href="http://dbpedia.org/resource/Hope">http://dbpedia.org/resource/Hope</a>	Tertiary	Esperanza	Contentment
Enthrallment	<a href="http://dbpedia.org/resource/Enthrallment">http://dbpedia.org/resource/Enthrallment</a>	Secondary	Embeleso	Joy
Enthrallment	<a href="http://dbpedia.org/resource/Enthrallment">http://dbpedia.org/resource/Enthrallment</a>	Tertiary	Embeleso	Enthrallment
Rapture	<a href="http://dbpedia.org/resource/Rapture">http://dbpedia.org/resource/Rapture</a>	Tertiary	Rapto/Arrebato	Enthrallment
Relief	<a href="http://dbpedia.org/resource/Relief">http://dbpedia.org/resource/Relief</a>	Secondary	Alivio	Joy

Relief	<a href="http://dbpedia.org/resource/Relief">http://dbpedia.org/resource/Relief</a>	Tertiary	Alivio	Relief
Surprise	<a href="http://dbpedia.org/resource/Surprise_(emotion)">http://dbpedia.org/resource/Surprise_(emotion)</a>	Primary	Sorpresa	
Amazement	<a href="http://dbpedia.org/resource/Amazement">http://dbpedia.org/resource/Amazement</a>	Tertiary	Asombro	Surprise
Astonishment		Tertiary	Asombro/Estupor	Surprise
Anger	<a href="http://dbpedia.org/resource/Anger">http://dbpedia.org/resource/Anger</a>	Primary	Ira	
Irritability	<a href="http://dbpedia.org/resource/Irritability">http://dbpedia.org/resource/Irritability</a>	Secondary	Irritabilidad	Anger
Aggravation		Tertiary	Agravación	Irritability
Agitation		Tertiary	Agitación	Irritability
Annoyance	<a href="http://dbpedia.org/resource/Annoyance">http://dbpedia.org/resource/Annoyance</a>	Tertiary	Molestia	Irritability
Grouchy		Tertiary	Malhumorado	Irritability
Grumpy		Tertiary	Gruñón	Irritability
Crosspatch		Tertiary	Malhumorado/Mal Genio	Irritability
Exasperation	<a href="http://dbpedia.org/resource/Exasperation">http://dbpedia.org/resource/Exasperation</a>	Secondary	Exasperación	Anger
Frustration	<a href="http://dbpedia.org/resource/Frustration">http://dbpedia.org/resource/Frustration</a>	Tertiary	Frustración	Exasperation
Rage	<a href="http://dbpedia.org/resource/Rage_(emotion)">http://dbpedia.org/resource/Rage_(emotion)</a>	Secondary	Rabia	Anger
Anger	<a href="http://dbpedia.org/resource/Anger">http://dbpedia.org/resource/Anger</a>	Tertiary	Ira	Rage
Outrage	<a href="http://dbpedia.org/resource/Outrage_(emotion)">http://dbpedia.org/resource/Outrage_(emotion)</a>	Tertiary	Atropello	Rage
Fury		Tertiary	Furia	Rage
Wrath		Tertiary	Ira/Colera	Rage
Hostility	<a href="http://dbpedia.org/resource/Hostility">http://dbpedia.org/resource/Hostility</a>	Tertiary	Hostilidad	Rage
Ferocity		Tertiary	Ferocidad	Rage
Bitter		Tertiary	Amargado	Rage
Hatred	<a href="http://dbpedia.org/resource/Hatred">http://dbpedia.org/resource/Hatred</a>	Tertiary	Odio	Rage
Scorn		Tertiary	Desdén	Rage
Spite		Tertiary	Despecho	Rage
Vengefulness	<a href="http://dbpedia.org/resource/Revenge">http://dbpedia.org/resource/Revenge</a>	Tertiary	Vengatividad	Rage
Dislike		Tertiary	Aversión	Rage
Resentment	<a href="http://dbpedia.org/resource/Resentment">http://dbpedia.org/resource/Resentment</a>	Tertiary	Resentimiento	Rage

Disgust	<a href="http://dbpedia.org/resource/Disgust">http://dbpedia.org/resource/Disgust</a>	Secondary	Disgusto	Anger
Revulsion		Tertiary	Repugnancia	Disgust
Contempt	<a href="http://dbpedia.org/resource/Contempt">http://dbpedia.org/resource/Contempt</a>	Tertiary	Desprecio	Disgust
Loathing		Tertiary	Asco	Disgust
Envy	<a href="http://dbpedia.org/resource/Envy">http://dbpedia.org/resource/Envy</a>	Secondary	Envidia	Anger
Jealousy	<a href="http://dbpedia.org/resource/Jealousy">http://dbpedia.org/resource/Jealousy</a>	Tertiary	Celos	Envy
Torment		Secondary	Tormento	Anger
Torment		Tertiary	Tormento	Torment
Sadness	<a href="http://dbpedia.org/resource/Sadness">http://dbpedia.org/resource/Sadness</a>	Primary	Tristeza	
Suffering	<a href="http://dbpedia.org/resource/Suffering">http://dbpedia.org/resource/Suffering</a>	Secondary	Sufrimiento	Sadness
Agony		Tertiary	Agonía	Suffering
Anguish	<a href="http://dbpedia.org/resource/Anguish">http://dbpedia.org/resource/Anguish</a>	Tertiary	Angustia	Suffering
Sadness	<a href="http://dbpedia.org/resource/Sadness">http://dbpedia.org/resource/Sadness</a>	Secondary	Tristeza	Sadness
Hurt		Tertiary	Daño	Sadness
Depression	<a href="http://dbpedia.org/resource/Depression_(mood)">http://dbpedia.org/resource/Depression_(mood)</a>	Tertiary	Depresión	Sadness
Despair		Tertiary	Desesperación	Sadness
Gloom	<a href="http://dbpedia.org/resource/Gloom">http://dbpedia.org/resource/Gloom</a>	Tertiary	Penumbra	Sadness
Glumness		Tertiary	Sombrío	Sadness
Unhappy		Tertiary	Infeliz	Sadness
Grief	<a href="http://dbpedia.org/resource/Grief">http://dbpedia.org/resource/Grief</a>	Tertiary	Dolor	Sadness
Sorrow	<a href="http://dbpedia.org/resource/Sorrow_(emotion)">http://dbpedia.org/resource/Sorrow_(emotion)</a>	Tertiary	Dolor/Pena	Sadness
Woe		Tertiary	Aflicción	Sadness
Misery		Tertiary	Miseria	Sadness
Melancholy	<a href="http://dbpedia.org/resource/Melancholy">http://dbpedia.org/resource/Melancholy</a>	Tertiary	Melancolía	Sadness
Disappointment	<a href="http://dbpedia.org/resource/Disappointment">http://dbpedia.org/resource/Disappointment</a>	Secondary	Decepción	Sadness
Dismay	<a href="http://dbpedia.org/resource/Dismay">http://dbpedia.org/resource/Dismay</a>	Tertiary	Consternación	Disappointment
Displeasure		Tertiary	Desagrado	Disappointment
Shame	<a href="http://dbpedia.org/resource/Shame">http://dbpedia.org/resource/Shame</a>	Secondary	Vergüenza	Sadness
Guilt		Tertiary	Culpa	Shame

Regret		Tertiary	Lamentar	Shame
Remorse	<a href="http://dbpedia.org/resource/Remorse">http://dbpedia.org/resource/Remorse</a>	Tertiary	Remordimiento	Shame
Neglect	<a href="http://dbpedia.org/resource/Neglect">http://dbpedia.org/resource/Neglect</a>	Secondary	Negligencia	Sadness
Alienation	<a href="http://dbpedia.org/resource/Alienation">http://dbpedia.org/resource/Alienation</a>	Tertiary	Alienación	Neglect
Defeatism	<a href="http://dbpedia.org/resource/Defeatism">http://dbpedia.org/resource/Defeatism</a>	Tertiary	Derrotismo	Neglect
Dejection		Tertiary	Abatimiento	Neglect
Embarrassment	<a href="http://dbpedia.org/resource/Embarrassment">http://dbpedia.org/resource/Embarrassment</a>	Tertiary	Vergüenza/Bochorno	Neglect
Homesickness	<a href="http://dbpedia.org/resource/Homesickness">http://dbpedia.org/resource/Homesickness</a>	Tertiary	Nostalgia	Neglect
Humiliation	<a href="http://dbpedia.org/resource/Humiliation">http://dbpedia.org/resource/Humiliation</a>	Tertiary	Humillación	Neglect
Insecurity		Tertiary	Inseguridad	Neglect
Insult	<a href="http://dbpedia.org/resource/Insult">http://dbpedia.org/resource/Insult</a>	Tertiary	Insult	Neglect
Isolation	<a href="http://dbpedia.org/resource/Social_isolation">http://dbpedia.org/resource/Social_isolation</a>	Tertiary	Aislamiento	Neglect
Loneliness	<a href="http://dbpedia.org/resource/Loneliness">http://dbpedia.org/resource/Loneliness</a>	Tertiary	Soledad	Neglect
Rejection	<a href="http://dbpedia.org/resource/Rejection">http://dbpedia.org/resource/Rejection</a>	Tertiary	Rechazo	Neglect
Sympathy	<a href="http://dbpedia.org/resource/Sympathy">http://dbpedia.org/resource/Sympathy</a>	Secondary	Simpatía	Sadness
Pity	<a href="http://dbpedia.org/resource/Pity">http://dbpedia.org/resource/Pity</a>	Tertiary	Piedad	Sympathy
Sympathy	<a href="http://dbpedia.org/resource/Sympathy">http://dbpedia.org/resource/Sympathy</a>	Tertiary	Simpatía	Sympathy
Fear	<a href="http://dbpedia.org/resource/Fear">http://dbpedia.org/resource/Fear</a>	Primary	Miedo	
Horror		Secondary	Horror	Fear
Alarm		Tertiary	Alarma/Susto	Horror
Shock		Tertiary	Escandalizar	Horror
Fear	<a href="http://dbpedia.org/resource/Fear">http://dbpedia.org/resource/Fear</a>	Tertiary	Miedo	Horror
Fright		Tertiary	Susto	Horror
Horror		Tertiary	Horror	Horror
Terror	<a href="http://dbpedia.org/resource/Terror">http://dbpedia.org/resource/Terror</a>	Tertiary	Terror	Horror
Panic	<a href="http://dbpedia.org/resource/Panic">http://dbpedia.org/resource/Panic</a>	Tertiary	Pánico	Horror
Hysteria	<a href="http://dbpedia.org/resource/Hysteria">http://dbpedia.org/resource/Hysteria</a>	Tertiary	Histeria	Horror
Mortification		Tertiary	Mortificación	Horror
Nervousness	<a href="http://dbpedia.org/resource/Nervousness">http://dbpedia.org/resource/Nervousness</a>	Secondary	Nerviosismo	

Anxiety	<a href="http://dbpedia.org/resource/Anxiety">http://dbpedia.org/resource/Anxiety</a>	Tertiary	Ansiedad	Nervousness
Suspense	<a href="http://dbpedia.org/resource/Suspense">http://dbpedia.org/resource/Suspense</a>	Tertiary	Incertidumbre/Suspense	Nervousness
Uneasiness		Tertiary	Inquietud	Nervousness
Apprehension	<a href="http://dbpedia.org/resource/Apprehension">http://dbpedia.org/resource/Apprehension</a>	Tertiary	Aprensión	Nervousness
Worry	<a href="http://dbpedia.org/resource/Worry">http://dbpedia.org/resource/Worry</a>	Tertiary	Preocupación	Nervousness
Distress	<a href="http://dbpedia.org/resource/Distress">http://dbpedia.org/resource/Distress</a>	Tertiary	Angustia/Dolor	Nervousness
Dread		Tertiary	Pavor	Nervousness

## 5. Anexo E

# Un Marco de Trabajo para Inteligencia Monitoreo e Interpretación de Comportamiento mediante Fusión de Información con Múltiples Sensores

(Informe descriptivo)

### Introducción

Cualquier tipo de avance en el campo de la inteligencia ambiental debe ser adaptativo por toda la gama de sensores que actualmente existen en el mercado; además sus aplicaciones deben tener un grado de acoplamiento muy bajo para permitir la conexión con otros sistemas y con diversos tipos de interfaces de salida; los marco de trabajo de monitoreo actuales tienen la capacidad de manejar todo tipo de redes de sensores, y con el avance y experiencia de usuario generado por los canales actuales de distribución de software han permitido el crecimiento de estos sistemas utilizando las opciones más efectivas disponibles en el mercado en cuanto tiene que ver con algoritmos de seguimiento o Tracking; contando con los beneficios brindados por el avance de la tecnología, resulta indispensable el desarrollo en líneas de interpretación de comportamientos complejos y análisis de situación para fusionarlos en aplicaciones que sean capaces de cubrir situaciones de monitoreo y vigilancia complejos más allá de lo que podría resultar evidente en un aplicativo convencional. La fusión de la información de múltiples fuentes, provee a las nuevas metodologías de desarrollo de aplicaciones mayor robustez, tolerancia a fallos y escalabilidad; el resultado final tras la puesta en acción de este tipo de aplicaciones reducen el ruido y la incertidumbre que se mide normalmente en aplicaciones orientadas a la seguridad monitoreo y control que no están enfocados en los postulados definidos en la inteligencia ambiental.

Las aplicaciones de la inteligencia ambiental pueden ser diseñadas empleando arquitecturas basadas en múltiples capas, facilitando su mantenimiento y dotando a los sistemas de avanzados protocolos para la comunicación con humanos agentes y otros sistemas semejantes o que manejen las mismas definiciones. El dinamismo generado por el desarrollo de aplicaciones orientados a la seguridad de las personas requiere el desarrollo de profesionales con las habilidades, herramientas y conocimientos necesarios para la creación de estos sistemas su escalamiento y mantenimiento; (Castillo, Fernández, & López, 2011) menciona como ejemplo a PRIS-MATICA como ejemplo de un sistema de vigilancia que combina sensores de audio y cámaras de seguridad para responder a toda una gama de situaciones en las redes de transportación pública. Este sistema consiste en una red de procesamiento distribuido (CCTV, IP, cámaras y sensores de audio) conectados a un nodo

de procesamiento principal que obtiene y transmite datos de sistemas remotos y varias interfaces de usuario.

Posteriormente la mayoría de los requerimientos hacen referencia al desarrollo de mecanismos de para el seguimiento de humanos y el análisis de actividad mediante medios múltiples cámaras; este es uno de los requerimientos más complejos y de los más costosos en tiempo de desarrollo, ya que implica procesar video en tiempo real, extraer frames y analizar las imágenes contenidas; la información obtenida debe compararse con una base de conocimiento que pueda categorizar el comportamiento en la imagen para finalmente decidir si se debe lanzar una alerta o no. En el desarrollo de su investigación (Castillo, Fernández, & López, 2011) proponen ejemplos como un sistema desarrollado para el seguimiento y clasificación de jugadores de futbol compuesto por ocho cámaras cada una provista con su propia unidad de procesamiento. Un sistema EasyLiving para el monitoreo en el contexto de la seguridad del hogar en el que un sistema de seguimiento con varias cámaras es usado para el reconocimiento de comportamiento en hogares. Finalmente proponen un caso aplicado en campo abierto donde un sistema de sensores heterogéneos para el seguimiento multimodal de vehículos en ambientes de interacción con humanos.

En su trabajo (Castillo, Fernández, & López, 2011) proponen un marco de trabajo de monitoreo e interpretación, apoyados en el paradigma modelo-vista-controlador con algunas modificaciones para poder adaptarla a los sistemas inteligentes distribuidos, generando una arquitectura con un nodo principal conectado a múltiples nodos remotos, donde el nodo remoto está encargado de la adquisición de datos y el procesamiento a bajo nivel como la segmentación y el seguimiento; mientras que el nodo central recibe toda la data y realiza el procesamiento de alto nivel como la detección de actividad por ejemplo.

### **El paradigma M-V-C**

En el campo de la ingeniería de software el modelo vista controlador es un patrón de construcción de software que aísla la interfaz de usuarios de la lógica del negocio y el modelo de datos manteniendo independiente el desarrollo, pruebas y manejo; está compuesto por tres componentes principales:

**Modelo:** Encargado de la adquisición de los datos de la aplicación que pueden provenir de toda clase de orígenes de datos como bases de datos, registros de máquina, texto plano, xml, etc. Este componente del modelo se encarga del manejo y persistencia de las entidades registrando toda clase de actividades relacionadas en los logs del sistema; es ampliamente usado sobre todo en sistemas donde se requiere persistencia de los objetos y en sistemas altamente transaccionales para aplicaciones con bases de datos por ejemplo;

cuando se realizan modificaciones o cualquier clase de operación CRUD el modelo notifica a los demás componentes como la vista y el controlador. El modelo da soporte a las siguientes operaciones

- Crear(C)
- Leer (R)
- Actualizar (U)
- Borrar (D)

**Vista:** Se procura mantener lo más liviana posible y no contener nada relacionado a la lógica de la aplicación; es básicamente la forma de interactuar con el usuario; una vista termina siendo a pesar de sus limitaciones la representación de un objeto; y un objeto puede tener varias vistas, a esta característica se la conoce como interacción multimodal; una de las ventajas de mantener la vista desacoplada de la aplicación, genera la capacidad de reutilizar el modelo y el controlador para interactuar con el usuario en otro tipo de terminales he interfaces; así por ejemplo un modelo y un controlador puede ser el mismo para una vista de una aplicación web, de escritorio, móvil; y cualquier otro medio de interacción con el usuario que pueda generarse posteriormente.

**Controlador:** Recibe las entradas de la vista generadas por el usuario y lanza las acciones del modelo; las acciones de un controlador en algunas ocasiones suelen llamarse por agentes inteligentes especialmente en aplicaciones de IA; en ambientes que involucra sesiones de usuario, la concurrencia (grado de petición de un recurso) sobre un sistema genera un número excesivo de instancias de controlador que saturan los nodos de un sistema web por lo general.

### **Extendiendo el paradigma MVC**

Para poder emplear el paradigma modelo vista controlador (Castillo, Fernández, & López, 2011) proponen dotar a la definición tradicional de este patrón con más flexibilidad; no solo para poder dar soporte como marco de trabajo para aplicaciones de monitoreo en ambientes inteligentes sino también para permitir la incorporación e implementación de nuevos algoritmos sin demasiado coste en cambios del diseño inicialmente propuesto; para este propósito se separa la lógica de negocio del modelo; y se introduce una nueva unidad en el patrón de diseño administrada por el controlador, esta nueva unidad se llama algoritmo y su funcionalidad estará dada por los diferentes módulos que componen el marco de trabajo propuesto; logrando así que se puede integrar nuevas funcionalidades; otra finalidad de la modificación integrada es alentar a usuarios externos a introducir sus propias funcionalidades al marco de trabajo que no necesariamente sean desarrolladores de la



propuesta; cada nivel de la nueva arquitectura es una implementación de modelo vista controlador que va desde los niveles más bajos como los módulos de adquisición de datos o de comunicación hasta el más alto nivel como el monitoreo, seguimiento y análisis de comportamiento. Otra ventaja de realizar una extensión del patrón de diseño MVC es que ya es ampliamente conocido por los desarrolladores de diferentes ámbitos y diferentes lenguajes; lo que permitirá su difusión y aceptación; además de tener la característica de poder integrarse con mucha facilidad con otros sistemas funcionando sobre el mismo paradigma; esta capacidad además disminuirá los costes de desarrollo y permitirá el escalamiento de las aplicaciones ya existentes.

Tradicionalmente todas las aplicaciones basadas en modelo vista controlador poseen los tres elementos básicos conocidos; pero la propuesta de (Castillo, Fernández, & López, 2011) implica generar múltiples módulos operando sobre un MVC extendido con la unidad algoritmo; la razón de esto es que en el ámbito de aplicaciones de monitoreo para ambientes inteligentes se requiere múltiples nodos de procesamiento; y un solo nodo con MVC no puede dar solución al procesamiento en tiempo real; varios módulos incomunicados o comunicados pero sin una integración remota; no permitirían la toma de decisiones en función de todas las entradas y sensores que un sistema Aml posee. Cada nodo de procesamiento del sistema envía datos a un módulo centralizado que considera todas las entradas del sistema distribuido y actúa como un backbone integrando todos los módulos de adquisición de datos sensorial y como administrados de acciones en función de determinado escenario de eventos; si un nodo requiere información adicional del modelo solo podrá obtenerla a través del nodo central y la comunicación entre nodos está dada únicamente a través del nodo central.

El nodo principal hace las veces de columna vertebral administrando toda la información de todos los nodos de un gran sistema nervioso de módulos y sensores del marco de trabajo propuesto; además reduce la dependencia que se tiene entre cada módulo; a través de entradas y salidas bien definidas; además se crea una vista común para contener todas las vistas de los módulos y la vista de cada módulo provee al usuario una interface para manipular los parámetros de configuración y monitorear el estado propio de cada módulo. El módulo principal posee también un controlador para monitorear el estado de cada módulo de adquisición de datos sensoriales.

Se debe considerar además que la comunicación con cualquier otro sistema externo independientemente del paradigma sobre el cual está desarrollado debe realizarse a través del módulo principal para garantizar la independencia de los módulos sensoriales; esta extensión puede exponer reportes de estado interfaces ya implementadas con otras

aplicaciones, sistemas de notificación espejos del propio sistema de procesamiento o cualquier otro agente capaz de generar información relevante al proceso de decisión en un Aml.

## **El marco de trabajo MVC propuesto**

### **Ejecución del modelo**

Previo a la descripción de cada nivel de marco de trabajo (Castillo, Fernández, & López, 2011) realiza un descripción de cómo se ejecuta el modelo propuesto; en primer lugar hay q tener en cuenta que el esquema es un modelo distribuido híbrido que sigue que sigue un esquema distribuido, donde cada nodo sensorial realiza las tareas de bajo procesamiento como por ejemplo leer las entradas de los sensores; auto administrar sus estados de activo e inactivo; cambiar los parámetros de adquisición del sensor; solicitar inhabilitación de la interface pedir reconexión al nodo principal y notificar anomalías con el correcto funcionamiento del sensor son algunas de las tareas que los nodos principales deberían realizar; El nodo central recibirá toda la data enviada por los módulos de adquisición de datos y fusionarlos para realizar las tareas que involucran procesamiento excesivo; el nodo remoto posee una estructura extendida del patrón MVC tradicional; el controlador global y la vista común también posee una estructura extendida y están alojadas en el nodo principal.

### **Capas del Framework**

Siendo el modelo basado en modelo vista controlador hay que hacer una diferenciación de la implementación del nodo central y los nodos remotos ya que ambos nodos utilizan la definición extendida propuesta; esta diferencia radica básicamente en los módulos contenidos por cada nodo; el punto más importante del framework extendido es el modelo común que es accesible por todos los módulos del sistema. Hay que notar que los módulos descritos no tienen una implementación específica para la unidad de algoritmo; esto responde a la necesidad de dejarlo genérico para poder extender el framework o componerlo según los requerimientos; continuación se resume los módulos contenidos por el nodo central y los nodos remotos:

- Nodos Remotos
  - Acquisition: Nivel de adquisición que interactúa directamente con los medios analógicos de adquisición de datos de los sensores; las medidas de datos de los sensores también podrían venir de cualquier medio como bases de datos el módulo de adquisición también debe ser capaz de interactuar con información pre procesada.

- Sensor Fusion: Sensor de fusión que mezcla la información obtenida por el sensor y la completa con información cualitativa para completar el análisis; Los algoritmos de fusión operan con diferentes espectros de imagen y entre dominios de conocimiento.
- Pixel-level Segmentation: Segmentación a nivel de pixel se dedicada principalmente a extraer objetos de interés de las imágenes procesadas el rango de operaciones referentes a esta capa puede ir desde procesos de binarización en infrarrojo IR (Fernández, Castillo, & Rodríguez, 2010) hasta operaciones más complejas con resultados más significativos.
- Pixel-level Segmentation Fusion: Fusión de segmentación a nivel de píxeles maneja todas las imágenes obtenidas son agrupadas según su localización; el framework deberá soportar más de un tipo de imágenes no solo IR.
- Blob Detection: Detección de manchas es un filtro a nivel de manchas en la imagen que no está en los niveles anteriores; específicamente la detección blob se encarga de la obtención de manchas y objetos permitidos mediante un análisis de objetos mucho más avanzado que en los módulos predecesores.
- Object Identification: La identificación de objetos opera mediante estructuras conocidas como manchas, esta información es abstraída y mapeada a objetos con coordenadas en el mundo real, no solo dentro de la imagen.
- Object Classification: La clasificación de objetos es de vital importancia para logra un buen análisis de actividad ya que provee información acerca de que es el objeto y su orientación respecto espacio físico que lo contiene,
- Object Tracking: El seguimiento de objetos realiza una identificación de las coordenadas de los objetos en el mundo real, los cálculos de trayectorias que pueden tener los objetos en el mundo real sin importar que clase de sensor registre el objeto; este nivel también hace posible la predicciones de trayectorias y entre otras cosas algún análisis básico de las trayectorias previas de un objeto. La información para el análisis a este nivel es obtenida del modelo común referente al mapa donde los sensores poseen rangos diferentes de observación.
- Event Detection: El nivel de detección de eventos analiza el comportamiento del objeto para encontrar información semántica; mediante las imágenes obtenidas y los datos relacionados de las capas anteriores se busca reconocer estados como cayendo, corriendo, caminando; la información generada a través de las capas es enviada al nodo central.

- **Nodo Central**
  - **Event Fusion:** El nivel de Fusión de eventos del nodo central se reciben datos desde todos los sensores para determinar áreas con situaciones de potencial interés; también se entrega la información global referente al espacio físico en monitores a cada nodo.
  - **Activity Detection:** La detección de actividades se ocupa de la relación de los estados que cada nodo puede dar respecto de las trayectorias globales; es decir que si se reconoce la actividad sospechosa en más de un nodo remoto el nodo central deberá relacionar la trayectoria y administrar la mejor decisión para el escenario; así por ejemplo si en dos nodos una persona se detectó corriendo y en tercero se detectó sentada habría que considerar que clase de acción tomar si es q el escenario lo amerita.

En un marco de trabajo para monitoreo e interpretación con múltiples sensores con sus respectivas interfaces de entrada orientado al monitoreo de un escenario común los eventos que provienen de orígenes distintos en la mayoría de los casos no coinciden, haciendo más difícil el proceso de análisis de actividad, esta es la razón principal por la que el marco de trabajo propuesto requiere un nivel de fusión de eventos; para unificar la información obtenida desde los diferentes sensores de adquisición de datos a través de los niveles previos. El último nivel del marco de trabajo es la detección de actividad, está encargado de la detección de actividades relacionadas a un mismo espacio físico o agrupado mediante algún criterio de división espacial en un rango muy específico de tiempo. La fusión de eventos en este punto posee un conocimiento más profundo de lo que está sucediendo realmente en un escenario; dotando al sistema de un alto nivel de abstracción.

### **Definición del modelo común**

Se introdujo la capa modelo común al marco de trabajo para poder dar respuesta el problema de intercambio de información entre los sensores remotos; la capa obtiene la información de todas las capas de los nodos remotos y puede proveer información primitiva respecto cada nivel de un nodo remoto, el modelo común es una variación del modelo del patrón tradicional MVC, de hecho el modelo común puede verse como un contenedor de información para todos los módulos remotos que la requieran, para este propósito se debe definir las directrices primitivas para dar soporte y administración de los datos obtenidos; para una correcta definición del marco de trabajo se empezó definiendo el trabajo realizado por cada capa de los nodos; dado que los parámetros de entrada y de salida en el escenario son conocidos; es posible estimar cuáles de ellos pertenecen al modelo común.