



Universidad Nacional de Educación a Distancia
Escuela Técnica Superior de Ingeniería Informática
Departamento de Inteligencia Artificial

Arquitectura para Sistemas Inteligentes Afectivos en Robots Tutores

Trabajo Fin de Máster

Directores: Dra. Ángeles Manjarrés Riesco y Dr. Félix De La Paz López

Luis Eduardo Imbernón Cuadrado

Madrid 2014 - 2015

Agradecimientos

Quiero agradecer el apoyo, la paciencia, la implicación personal y la gran cantidad de tiempo dedicado de Ángeles y de Félix, directores de este trabajo y excelentes personas. Sin ellos, realizar este trabajo hubiera sido imposible.

Al colegio Los Peñascales de Las Rozas (Madrid), y concretamente a Mónica, profesora de primaria que me ayudó durante las sesiones de grabación, así como a los padres voluntarios que tomaron notas durante las sesiones para esta investigación.

A Juan y a Ramón por su amistad incondicional durante casi la mitad de mi vida. Por sus consejos sabios y por sus consejos disparatados, pero sobre todo, por su presencia en los mejores y en los peores momentos. Existen pocas anécdotas en mi vida en las que ellos no aparezcan.

A mis padres Charo y Jose Luis por ser una referencia para mí, y que gracias a su lucha, su educación, su apoyo y su cariño he llegado hasta donde estoy. Ellos han conseguido que el camino sea más fácil. Sin ellos, nada de esto hubiera sido posible.

A mis hermanos Alfredo y Jose Antonio por sus ánimos, por los momentos cómicos alrededor de la mesa, por sus cuidados, por llenar mi memoria de buenos recuerdos y por todo su cariño.

A mi familia en general, pero especialmente a mis tíos Chema, Eduardo, Mariló y Pilar, porque sé que cuento con ellos para lo que haga falta.

Por último a Sherezade, por su paciencia, su amor y por todos los detalles del día a día. Porque ha conseguido hacerme sonreír todos los días y porque ha sabido estar en los peores momentos.

Este trabajo supone el inicio de un camino me apasiona y que siempre quise emprender.

"Nada grande se ha hecho en el mundo sin una gran pasión". Georg Wilhelm Friedrich Hegel

Tabla de contenidos

Acrónimos y otras Abreviaciones	1
Resumen	4
Abstract	5
1.Introducción	8
1.1. Motivación	8
1.2. Contexto de la investigación	9
1.3. Definición del problema e Hipótesis del trabajo	10
1.4. Objetivos y Metodología	10
1.5. Aportaciones	12
1.6. Organización de la memoria	12
2.Estado del Arte	16
2.1.Lenguajes de marcas	16
2.2.Reconocimiento de estados emocionales a partir de análisis de logs	22
2.3.Métodos de elicitación de conocimiento para el desarrollo de software educativo afectivo	27
2.4.Scratch en la educación	29
2.5.Robótica en educación	30
3.Arquitectura ASIART	38
3.1.Arquitectura general	38
4.Experiencia de usuario con Scratch	42
5.Metodología ASIART	47
6.Desarrollo del modelo emocional	50
6.1.Módulo de análisis de logs	50
6.1.Descripción de la muestra	52
6.2.Selección de atributos	53
6.3.Modelo de clasificación	61
6.4.Análisis de los resultados	65
7.Desarrollo del modelo de intervención pedagógica para la práctica con Scratch	72
8.Plataforma de desarrollo de robots tutores afectivos en el marco de la arquitectura ASIART	76
8.1.Cliente (Data Recovery)	76
8.2.Web Server	81
9.Implementación de ASIART con SCRATCH y NAO	84
10.Experiencia de tutorización	91
11.Evaluación de ASIART	95

12.Conclusiones	100
13.Trabajos Futuros	102
Anexo 1: Formulario basado en BROMPT	105
Anexo 2: WSDL del servicio web emocional	106
Anexo 3: WSDL del servicio web de lenguaje de etiquetado	107
Bibliografía	110

Índice de figuras

Figura 2.1: La plataforma Greta controlando los comportamientos de NAO (izquierda) y un agente virtual (derecha). (fuente: [Paiva et al., 2012])	18
Figura 2.2: Estructura de VHML. (fuente: [Beard et al., 2001]).	19
Figura 2.3: Módulo de conversación de MPML (fuente: [Mori et al., 2003]).	20
Figura 3.1: Patrón arquitectural de ASIART	38
Figura 4.1: Interfaz del software de registro de logs y vídeos	43
Figura 4.2: Fotograma de la experiencia de usuario con Scratch	44
Figura 6.1: Interfaz principal del software de etiquetado	51
Figura 6.2: Interfaz de reproducción del software de etiquetado	51
Figura 6.3: Interfaz del software de análisis de datos	52
Figura 6.4: Topología de un Clasificador Naïve Bayes	61
Figura 6.5: Topología de un perceptrón multicapa	62
Figura 6.6: Topología de la red neuronal utilizada para la clasificación	67
Figura 8.1: Diagrama de despliegue de la arquitectura ASIART	76
Figura 8.2: Interfaz principal de DataRecovery	77
Figura 8.3: Interfaz de configuración de DataRecovery	77
Figura 8.4: Diagrama de clases de las clases principales de DataRecovery	78
Figura 8.5: Diagrama de clases de los escuchadores de eventos de teclado y ratón de DataRecovery	79
Figura 8.6: Diagrama de clases del procesador de logs de DataRecovery	80
Figura 8.7: Diagrama de clases del post procesador de logs de DataRecovery	80
Figura 8.8: Diagrama de secuencia de DataRecovery desde la recogida de datos hasta el envío por servicio web	81
Figura 8.9: Diagrama de clases del servicio web emocional	82
Figura 8.10: Diagrama de secuencia del servicio web emocional	82
Figura 8.11: Diagrama de clases del servicio web del lenguaje de etiquetado	83
Figura 8.12: Diagrama de secuencia del servicio web del lenguaje de etiquetado y los generadores de modelos de intervención pedagógica y de estudiante	83
Figura 9.1: Arquitectura ASIART con la implementación de Scratch y NAO.	84
Figura 9.2: Diagrama de clases para la transformación de BML	85
Figura 9.3: Diagrama de clases para la traducción de BML a acciones de NAO	86
Figura 9.4: Diagrama de clases de la clase principal de la librería de traducción de BML a NAO.	87
Figura 9.5: Diagrama de secuencia del servicio web del lenguaje de etiquetado y la librería de traducción de BML a NAO.	88
Figura 10.1: Fotograma de la experiencia de tutorización	91

Índice de tablas

Figura 4.3: Fotograma de la experiencia de usuario con Scratch	44
Tabla 6.1: Datos de muestra por estado de concentración	52
Tabla 6.2: Atributos de entrada para los métodos de selección y evaluación	53
Tabla 6.3: Tabla de resultados de la selección y evaluación de atributos	60
Tabla 6.4: Selección de atributos de la muestra de datos	60
Tabla 6.5: Matriz de confusión del algoritmo C4.5	65
Tabla 6.6: Valores AUC/ROC obtenidos en C4.5	65
Tabla 6.7: Matriz de confusión del algoritmo KNN	66
Tabla 6.8: Valores AUC/ROC obtenidos en KNN	66
Tabla 6.9: Matriz de confusión del algoritmo de redes neuronales	67
Tabla 6.10: Valores AUC/ROC obtenidos en la red neuronal	67
Tabla 6.11: Matriz de confusión del algoritmo SVM	68
Tabla 6.12: Valores AUC/ROC obtenidos en SVM	68
Tabla 6.13: Matriz de confusión del algoritmo Naive Bayes	68
Tabla 6.14: Valores AUC/ROC obtenidos en Naive Bayes	68
Tabla 7.1: Tipo de tutor según la competencia y motivación del aula (fuente: [Hernando, 2015]).	72
Tabla 9.1: Respuestas de los alumnos a la sección 1 de la evaluación	95
Tabla 9.2: Respuestas de los alumnos a la sección 2 de la evaluación	95
Tabla 9.3: Respuestas de los alumnos a la sección 3 de la evaluación	96
Tabla 9.4: Respuestas de los alumnos en la sección 4 de la evaluación	96
Tabla 9.5: Respuestas de los alumnos en la sección 5 de la evaluación	96

Índice de algoritmos

Algoritmo 6.1 C4.5	54
Algoritmo 6.2 K2	56
Algoritmo 6.3 ReliefF	59
Algoritmo 6.4 Retropropagación	62
Algoritmo 6.5 K-vecinos	64

Acrónimos y otras Abreviaciones

Acrónimo / Abreviación	Definición
AIML	Artificial Intelligence Markup Language
A.L.I.C.E	Artificial Linguistic Internet Computer Entity Chatterbot
ARFF	Attribute-Relation File Format
ASD	Autism Spectrum Disorder
ASIART	Arquitectura para Sistemas Inteligentes Afectivos en Robots Tutores
AUC	Area Under The Curve
BAML	Body Animation Markup Language
BFO	Basic Formal Ontology
BML	Behavior Markup Language
BROMPT	Baker-Rodrigo Ocumpaugh Monitoring Protocol
CFS	Correlation Feature Selection
CLASS	Classroom Assessment Scoring System
CNTK	Computational Network Toolkit
DMML	Dialogue Manager Markup Language
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DTD	Document Type Definition
ECA	Embodied Conversational Agent
EM	Expectation Maximization
EML	Emotional Markup Language
EmotionML	Emotion Markup Language
EMO	Emotion Ontology
FAML	Facial Animation Markup Language
FML	Functional Markup Language
FOAF	Friend Of A Friend
FSLSM	Felder-Silverman Learning Style Model
GUMO	General User Model Ontology
GML	Gesture Markup Language
HART	Human Affect Recording Tool
HEO	Human Emotion Ontology

Acrónimo / Abreviación	Definición
HRI	Human-Robot Interface
IG	Information Gain
IHM	Interfaz Hombre Máquina
IP	Internet Protocol
KAR	Kindergarten Assistive Robotics
KNN	K Nearest Neighbors
MFO	Mental Functioning Ontology
MFO-EM	Mental Functioning Ontology - Emotion Ontology
MIT	Massachussetts Institute of Technology
MLP	Multilayer Perceptron
MQI	Mathematical Quality of Instruction
NB	Naïve Bayes
NLP	Natural Language Processing
NTP	Network Time Protocol
OMD	Ontology of Mental Disease
OUR-K	Ontology-based Unified Robot Knowledge Framework
OWL	Web Ontology Language
PML	Perception Markup Language
RBF	Radial Basis Function
RoboML	Robotic Markup Language
ROC	Receiver Operating Characteristic
SAIBA	Situation, Agent, Intention, Behavior, Animation
SAM	Self-Assessment Manikin
SAR	Socially Assistive Robotics
SML	Speech Markup Language
SOAP	Simple Object Access Protocol
STT	Speech To Text
SVM	Support Vector Machines
TCP	Transmission Control Protocol
TH	Talking Head
TTS	Text To Speech
VHML	Virtual Human Markup Language

Acrónimo / Abreviación	Definición
VLE	Virtual Learning Environment
W3C	World Wide Web Consortium
xDSML	eXecutable Domain Specific Modeling Languages
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

Resumen

Durante la última década la robótica ha atraído un gran interés de los profesores e investigadores como una herramienta valiosa para desarrollar habilidades cognitivas y sociales de los estudiantes, desde el nivel preescolar hasta las enseñanzas medias, dando apoyo principalmente en ciencias, tecnología y actividades de aprendizaje interdisciplinar. La implementación de comportamientos de apoyo social en robots tutores, y más concretamente en la dimensión afectiva, puede resultar una contribución significativa a la eficiencia del aprendizaje, particularmente en los primeros ciclos educativos.

Con el objetivo de contribuir al floreciente campo de los robots tutores afectivos en la educación primaria, en este proyecto se ha propuesto un patrón arquitectural que integra cualquier software educativo destinado a niños de primaria con un componente cuya función es la identificación del estado afectivo de los estudiantes que interactúan con el software (mediante el análisis de las interacciones del teclado y el ratón) y con el controlador de un “robot tutor” que proporciona soporte pedagógico “afectivo” a estos estudiantes.

Para apoyar la implementación de robots tutores afectivos conforme a la arquitectura propuesta, se han desarrollado además una metodología, que incorpora un método de elicitación de conocimiento pedagógico, y una plataforma genérica. La plataforma contiene un módulo para la identificación de estados emocionales basado en el método Naïve Bayes, y un módulo genérico (un *framework*) para el soporte pedagógico afectivo. Este componente pedagógico modela las intervenciones afectivas de un robot (incluyendo las expresiones faciales, lenguaje corporal, tono de voz...) en términos de ficheros BML (Behavior Model Language) como función del estado emocional de los estudiantes, e incluye asimismo un traductor de dichas intervenciones a las acciones del robot. De este modo, se posibilita dotar al robot tutor de las características sociales de los agentes virtuales que pueden especificarse mediante BML. Tanto la plataforma como la metodología están adaptadas a la idiosincrasia de los alumnos de primaria. Finalmente, se ha ilustrado el uso de esta plataforma implementando un prototipo de la arquitectura donde el software educativo fue instanciado con Scratch y el robot tutor con NAO.

De cara a orientar el desarrollo de la plataforma y del prototipo, se ha realizado una experiencia de usuario que consistió en dos sesiones de 45 minutos con un grupo de 10 estudiantes con edades comprendidas entre los 9 y 10 años, en donde se recopilaban los datos de las interacciones de teclado y ratón mientras los estudiantes utilizaban el software educativo Scratch tutorizados por su profesora. Las sesiones fueron también grabadas en videos para analizar posteriormente las respuestas de la profesora en los diferentes “escenarios afectivos”.

En esta investigación se ha concluido que es posible identificar las emociones que condicionan de forma más relevante el carácter de la intervención educativa, para el caso de alumnos de primaria, sin utilizar métodos intrusivos y de alto coste. También se ha concluido que es posible concebir una arquitectura de componentes desacoplados de propósito general, donde se pueden integrar un amplio rango de herramientas software educativas y robot tutores que actúen de acuerdo a diferentes criterios pedagógicos.

Abstract

Over the last decade robotics has attracted high interest from teachers and researchers as a valuable tool for developing the cognitive and social skills of students, from preschool to high-school levels, supporting specific subjects such as science and technology, or supporting interdisciplinary learning activities. The implementation of social-support behaviors in tutoring robots, in particular in the emotional dimension, can make a significant contribution to learning efficiency.

With the aim of contributing to the rising field of affective tutor robotics, we propose an architectural pattern that integrates any given educational software for elementary school children with a component whose function is to identify the emotional state of the students that are interacting with the software by analyzing their keyboard and mouse interactions, and with the driver of a tutoring robot that provides personalized emotional pedagogical support to the students.

In order to support the affective tutoring robots implementation according to the proposed architecture, we have also developed a methodology that incorporates an elicitation method of pedagogical knowledge, and a generic platform. This platform contains a model for the identification of emotional states, founded on the Naïve Bayes method, and a generic affective pedagogical support model (a framework). This pedagogical component models the affective educational interventions (including facial expressions, body language, tone of voice,...) in terms of BML files (Behavior Model Language) in a way that is dependent on the emotional state of the students, and then translates the BML Markup Language to the actions of a tutoring robot. In this way, we provide to the tutoring robot the social characteristics of the virtual agents that could be specified through BML. Both the platform and the methodology are adapted to the primary school students. Finally, we have illustrated the use of this platform by using it to build a prototype implementation of the architecture, in which the educational software was instantiated with Scratch and the tutoring robot with NAO.

With the purpose to orient the development of the platform and of the prototype we carried out a user experiment consisting of two 45-minute sessions with a group of 10 students aged 9 and 10, in which we collected keyboard and mouse interaction data while students used the educational software. The sessions were also recorded on video to later analyze the responses of the teacher in different "affective scenarios".

We conclude from our work that it is possible to identify, without using intrusive identification methods, the emotions that most affect the character of educational interventions, at least in the case of elementary students. Our work also demonstrates the feasibility of a general-purpose architecture of decoupled components, in which a wide range of educational software and tutoring robots can be integrated and then used according to different educational criteria.

CAPÍTULO 1

INTRODUCCIÓN

Este capítulo es un resumen del resto de la memoria e introduce la investigación llevada a cabo en este trabajo fin de máster (presentando las motivaciones, problemas, aproximaciones, hipótesis, objetivos y metodologías) y expone la estructura del documento.

1. Introducción

Durante la última década la robótica ha atraído un gran interés de los profesores e investigadores como una herramienta valiosa para desarrollar habilidades cognitivas y sociales de los estudiantes, desde el nivel preescolar hasta las enseñanzas medias, dando apoyo en ciencias, matemáticas, tecnología, informática y otras materias o actividades de aprendizaje interdisciplinar [Alimisis, 2013].

La implementación de comportamientos de apoyo social en robots tutores incrementa la eficiencia en el aprendizaje de los estudiantes [Saerbeck et al., 2010]. Más específicamente, tomar en cuenta la dimensión afectiva del aprendizaje en el diseño de robots u otras interfaces de sistemas informáticos educativos para niños contribuye activamente al cumplimiento de los objetivos de aprendizaje.

La educación emocional ha de ser un proceso continuo y permanente que debe estar presente a en todo el currículum académico y en la formación permanente a lo largo de toda la vida. Entre los objetivos principales de la educación emocional se encuentran la comunicación de emociones y la identificación de las emociones de los demás [Bisquerra, 2005]. Uno de los requisitos principales de un robot tutor afectivo debe ser por tanto la capacidad de reconocimiento y expresión de emociones.

El fin último de este proyecto ha sido contribuir a la integración de robots tutores afectivos no intrusivos y de bajo coste en la enseñanza primaria. En los apartados siguientes, se detallan los fundamentos de este trabajo desde sus motivaciones hasta las aportaciones propuestas para el problema planteado. Finalmente, este capítulo de introducción termina exponiendo la organización de esta memoria.

1.1. Motivación

El auge de la web semántica y de los sistemas educativos de aprendizaje en línea ha motivado en la última década propuestas de estandarización de modelos de competencias, perfiles de estudiantes y objetos de aprendizaje entre otros, con el objetivo de representar el conocimiento contenido en los recursos web en forma disponible para las aplicaciones [Gascueña et al, 2004], particularmente mediante ontologías [Kaur and Chaudhary, 2015]. Así, en [Heckmann et al., 2007], [Gascueña et al., 2004], [Sieg et al., 2007] y [Beetz, 2013] se formalizan los mencionados modelos en ontologías como GUMO (General User Model Ontology) y OUR-K (Ontology-based Unified Robot Knowledge Framework), haciendo uso de OWL (Web Ontology Language) como lenguaje de marcado.

Dado el interés de los sistemas educativos afectivos, al igual que en el caso de los sistemas educativos de aprendizaje en línea, se ha motivado la formalización de los modelos de emociones en ontologías y lenguajes de marcado, como en el caso de [Bertola and Patti, 2013], [Grassi, 2009], [Hastings et al., 2012] y [Hastings et al., 2011] donde se formalizan los modelos en ontologías como HEO (Human Emotion Ontology), MFO-EM (Mental Functioning Ontology - Emotion Ontology), EMO (Emotion Ontology), BFO (Basic Formal Ontology) y OMD (Ontology of Mental Disease) haciendo uso asimismo de OWL como lenguaje de marcado.

La educación emocional adopta un enfoque del ciclo vital, que se lleva a la práctica a través de programas secuenciales que se inician en la educación infantil, siguen a través de primaria y secundaria y se prolongan en la vida adulta [Gascueña et al., 2004]. Tal como se describe en [Riggs et al., 2006], en edades tempranas el desarrollo emocional es previo

al desarrollo del pensamiento y debe ser integrado con las habilidades lingüísticas y cognitivas, más lentas de desarrollar. Otras singularidades de la educación primaria se ponen de manifiesto en estudios como [Duit and Treagust, 2013] donde se señalan diferencias con respecto a otros ciclos educativos como la incapacidad de extrapolar conceptos de un dominio a otro y la poca duración de la retención de estos conceptos.

A nuestro entender, en el ámbito de los sistemas educativos de aprendizaje en línea y los sistemas recomendadores educativos afectivos no se ha tomado en suficiente consideración la idiosincrasia de la educación primaria. En particular, no se encuentran referencias acerca de ontologías de emociones ni competencias específicas de alumnos de primaria, ni se han explotado las potencialidades de los robots tutores afectivos utilizando procedimientos de detección de estados emocionales de los educando no intrusivos y de bajo precio, de modo que puedan ser utilizados de forma habitual en los centros educativos.

Existen dos métodos para la detección de estados emocionales en estudiantes que interactúan con un software educativo: un método fisiológico y un método de comportamientos. En [Resch et al., 2015], [Picard et al., 2001] y [Bailenson et al., 2007] se reportan estudios sobre las emociones basados en parámetros fisiológicos medidos mediante dispositivos intrusivos como pulseras bluetooth para obtener las pulsaciones, la variación de las pulsaciones o la temperatura de la piel entre otros, o dispositivos hápticos. Estos dispositivos tienen el inconveniente de que no son baratos y de que, dependiendo de la información fisiológica que se desee leer, es posible que sea necesario el uso de más de un dispositivo a la vez, lo que puede ocasionar cierta incomodidad al alumno. Estos factores son especialmente relevantes en alumnos de corta edad.

Dado que los robots tutores actuales carecen de los datos de interacción con el ordenador de los que puede disponer un software educativo para diagnosticar estados afectivos (entradas de teclado y ratón, e imágenes faciales de fácil interpretación cuando el estudiante mira a la pantalla), no pueden exhibir una conducta social tan compleja como un agente virtual, a no ser que evalúen el estado afectivo de los alumnos por métodos intrusivos y de alto precio. Sin embargo, tal y como se concluye en [Saerbeck et al., 2010] existe una mejor interacción social de los alumnos con los robots que con los agentes virtuales.

En base a lo anteriormente expuesto, el trabajo presentado en esta memoria está motivado principalmente por el propósito de mejorar la robótica en educación primaria, dotándola de la potencialidad de dar respuestas pedagógicas afectivas en base a los métodos no intrusivos y de bajo coste de reconocimiento de emociones que poseen los sistemas recomendadores educativos afectivos. Una motivación secundaria pero también importante ha sido enriquecer los estándares educativos incorporando las peculiaridades de los niños de primaria.

1.2. Contexto de la investigación

Este trabajo se encuadra en dos marcos de investigación:

- Dentro del marco de los robots tutores afectivos, que ya han sido empleados en numerosos estudios (terapia con niños autistas [Ismail et al., 2012], clases de inglés en primaria [Keren et al., 2012], presentaciones de temarios en clases de primaria [Nozawa et al., 2004] , etc), y que tal como muestran los resultados de [Saerbeck et al., 2010],

mejoran la eficiencia del aprendizaje de los estudiantes, que aprenden más y estando más motivados.

- Dentro del marco de los sistemas recomendadores afectivos, que tal como se describe en [Salmeron-majadas, 2014] resultan de gran importancia dentro de la educación, debido a la fuerte relación existente entre las emociones y los procesos mentales de carácter cognitivo.

Las dimensiones de estas áreas que conciernen a nuestra investigación serán estudiadas más en profundidad en el capítulo dedicado al estado del arte.

1.3. Definición del problema e Hipótesis del trabajo

Con este trabajo se pretende facilitar la implementación de robots tutores afectivos que operen vinculados a un software educativo concebido para niños de primaria, tomando en cuenta la idiosincrasia de estos estudiantes. Tales robots habrán de desempeñar el papel de asesores virtuales en un sistema educativo afectivo, esto es, serán capaces de identificar el estado emocional de los estudiantes y dar una respuesta pedagógica acorde.

Este propósito se ha concretado en dos objetivos principales:

- Ofrecer una arquitectura de propósito general y una metodología y plataforma software para su implementación con un amplio rango de softwares educativos, bajo cualquier enfoque pedagógico y con un amplio rango de robot.
- Realizar aportaciones en el ámbito de los estándares educativos particularizados en los estudiantes de primaria.

Nuestras hipótesis de trabajo son las siguientes:

- Es posible concebir una arquitectura de componentes desacoplados de propósito general donde se puedan integrar un amplio rango de herramientas software educativas y robots que puedan operar de acuerdo a diferentes criterios pedagógicos.
- Es posible identificar las emociones que condicionan de forma más relevante el carácter de la intervención educativa, para el caso de alumnos de primaria, sin utilizar métodos intrusivos y de alto coste.
- Los estándares educativos actuales no están bien adaptados a la idiosincrasia de los alumnos de primaria, particularmente en lo que respecta a los modelos de competencias y estados cognitivo-afectivos que condicionen el carácter de las intervenciones pedagógicas.
- Es posible desarrollar un traductor estándar del comportamiento de un avatar virtual educativo estándar a un robot afectivo, con algunas simplificaciones.

1.4. Objetivos y Metodología

Los objetivos mencionados en la sección anterior implican la integración de las investigaciones en sistemas recomendadores afectivos y las investigaciones en robots tutores.

Para alcanzar dichos objetivos generales, se han planteado los siguientes objetivos y subobjetivos más concretos:

1) Realizar un estudio preliminar del estado de la tecnología en los múltiples ámbitos que atañen al proyecto, esto es, en lo referente a:

- La robótica afectiva dentro de la educación.
- Las metodologías de observación en ingeniería de requisitos.
- La identificación de estados emocionales mediante métodos no intrusivos.
- Los estándares educativos:
 - Ontologías del alumno.
 - Ontologías de emociones.
- Los lenguajes de marcado de robots y avatares virtuales.

2) Definir una arquitectura de propósito general que integre un software educativo destinado a niños de primaria con un componente cuya función es la identificación del estado afectivo de los estudiantes que interactúan con el software, y con el controlador de un "robot tutor" que proporciona soporte pedagógico "afectivo" a estos estudiantes.

3) Definir una metodología para la implementación de aplicaciones basadas en la arquitectura propuesta, que incluya un método de elicitación de conocimiento pedagógico basado en experiencias de usuario.

4) Desarrollar una plataforma genérica para dar soporte a la implementación de robots tutores afectivos conforme a la arquitectura propuesta.

5) Ilustrar el uso de la plataforma y la aplicación de la metodología definida utilizando el software educativo Scratch y el robot NAO.

Para cumplir con los objetivos 4 y 5 se han planteado los siguientes subobjetivos:

- Realizar sesiones de grabación de vídeo e interacciones de teclado y de ratón en una clase de programación con Scratch para alumnos de quinto de primaria (experiencia de usuario).
- Analizar los vídeos y registros de interacciones de teclado y de ratón para el etiquetado de los registros con estados emocionales.
- Analizar los vídeos para el diseño del modelo de intervención pedagógica.
- Desarrollar el prototipo y la plataforma, que incluirán un modelo emocional basado en métodos de minería de datos (en base a los registros etiquetados de teclado y ratón), un modelo de intervención pedagógica para la práctica con Scratch y un modelo de estudiante de primaria que contemple la dimensión afectiva, en base a ontologías que incorporen las especificidades de dichos estudiantes. Este desarrollo implicará:
 - Desarrollar una API multiplataforma para la recopilación de las interacciones de teclado y de ratón.
 - Implementar un servicio web de consulta al modelo emocional
 - Implementar un servicio web de consulta al modelo de intervención pedagógica.

- Implementar un sistema de traducción de lenguaje de marcado a instrucciones para el robot NAO.
 - Evaluar la solución propuesta en una experiencia conducida en una asignatura de un colegio de primaria
- 6) Analizar y discutir los resultados obtenidos, contrastando nuestras hipótesis de trabajo.

1.5. Aportaciones

Las aportaciones de este trabajo se derivan, por un lado del desarrollo de la solución al problema planteado, y por otro lado de cuestiones surgidas al establecer los requisitos de dicha solución.

Con ASIART se hacen las siguientes aportaciones al campo de los sistemas recomendadores educativos afectivos y los robots tutores afectivos:

- Una arquitectura, metodología y plataforma de desarrollo para la implementación de robots tutores afectivos que operen asociados a un software educativo, haciendo las veces de agentes virtuales afectivos en un sistema recomendador educativo.
- Un modelo de identificación de estados emocionales no intrusivo y de bajo coste, implementado mediante métodos de minería de datos, y basado en las interacciones de teclado y ratón.

Asimismo, se realizan aportaciones en el campo de la observación y la toma de notas en experiencias de usuario conducidas con alumnos de primaria, siendo estas:

- Extensión del protocolo de toma de notas BROMP, llevándolo a la toma de notas manual (sin dispositivos móviles).
- Metodología de grabación y recopilación de datos de interacción con teclado y ratón.

1.6. Organización de la memoria

Esta memoria se organiza en trece capítulos acompañados de tres anexos, los agradecimientos y la bibliografía:

1. Introducción. Capítulo actual, donde se describen las motivaciones, el contexto, el problema planteado, las hipótesis, los objetivos, y la metodología, así como las aportaciones de este proyecto.
2. Estado del Arte. Capítulo dedicado a la revisión de los trabajos más relevantes para los propósitos de este proyecto. Se dedica una sección a cada una de las áreas de interés.

3. Arquitectura ASIART. Capítulo en que se describe el patrón arquitectural propuesto para la implementación de robots tutores afectivos que operen asociados a un software educativo.
4. Experiencia de usuario con Scratch. Capítulo en que se detallan los procedimientos seguidos para obtener los datos que han servido para el desarrollo del modelo emocional y el modelo de intervención pedagógica.
5. Metodología ASIART. En este capítulo se detalla la metodología para la implementación de robots tutores afectivos con ASIART.
6. Desarrollo del modelo emocional. Capítulo en que se detallan los procesos de análisis de datos, desde la selección de las características más relevantes, hasta la creación del modelo emocional.
7. Desarrollo del modelo de intervención pedagógica para la práctica con Scratch. Capítulo en que se detalla el desarrollo del modelo de intervención pedagógica implementado en el prototipo propuesto.
8. Plataforma de desarrollo de robots tutores afectivos en el marco de la arquitectura ASIART. Capítulo en que se detallan las características de la plataforma de desarrollo y su funcionamiento.
9. Implementación de ASIART con SCRATCH y NAO. En este apartado se describe los componentes de la arquitectura que han sido adaptados para ejecutar las acciones en NAO y usando SCRATCH como software educativo.
10. Experiencia de tutorización. Capítulo en que se describen la preparación y la realización de una experiencia para una validación preliminar del robot tutor para la enseñanza con Scratch.
11. Evaluación de ASIART: después de describir la encuesta utilizada para recoger las opiniones de los alumnos, se exponen los resultados obtenidos, así como una discusión de los mismos.
12. Conclusiones
13. Trabajos futuros

Por otra parte, se adjuntan a esta memoria diversos archivos:

- Formulario BROMPT utilizado en las sesiones de grabación por los observadores.
- Descripciones de las interfaces de servicios web (WSDL).

CAPÍTULO 2

ESTADO DEL ARTE

Este capítulo contiene el estado del arte de todas las cuestiones relacionadas con esta investigación: los lenguajes de marcas, el reconocimiento de estados emocionales a partir de análisis de registros de interacciones de teclado y ratón, los métodos de elicitación del conocimiento para el desarrollo de software educativo afectivo, Scratch en la educación y a robótica en la educación.

2. Estado del Arte

La arquitectura propuesta en este proyecto, tal como se ha descrito en apartados anteriores, implica la identificación de los estados emocionales que suscita el uso de un sistema software educativo, así como la determinación de la respuesta pedagógica más acorde a cada estado emocional y perfil de alumno, en un contexto dado de aprendizaje, y la expresión de esta respuesta pedagógica a través de un robot tutor afectivo.

En la literatura, la identificación de estados emocionales se realiza principalmente mediante la implementación de métodos de minería de datos. Estos métodos, tal como se describe en [Hernández Orallo et al., 2004] se dividen en varias fases que pueden ser resumidas en: recopilación de los datos necesarios para el análisis, integración de estos datos en caso de provenir de diferentes fuentes, implementación del método de minería de datos y evaluación del método escogido. Este proyecto basa la identificación de los estados emocionales en el análisis de los registros (*logs*) de las interacciones de teclado y ratón. Puesto que no se dispone de una base de datos que relacione las interacciones de teclado y ratón con estados emocionales determinados (ni en el caso concreto de alumnos de primaria ni en ningún otro caso) no ha sido necesaria una fase de integración de datos pero sí una fase de recopilación de datos que se ha realizado mediante métodos de observación en ingeniería de requisitos. Por lo tanto se ha llevado a cabo un estudio del estado del arte de los métodos de observación en ingeniería de requisitos existentes, así como de los métodos de minería de datos implementados para la clasificación de estados emocionales basados en *logs* de interacciones de teclado y ratón.

Para la identificación de los estados emocionales motivados por el uso de un software educativo, en este proyecto hemos experimentado con Scratch. En [Bredenkamp and Copple, 2009] se comenta la amplia divulgación de Scratch en estudios que van desde la enseñanza primaria hasta la enseñanza universitaria. En este capítulo se resume también un estudio realizado sobre el uso de Scratch en la educación, profundizando en el uso de esta herramienta dentro de la educación primaria.

Con ASIART se propone la integración de software educativo con un robot que haga las veces de un avatar virtual de un sistema recomendador educativo afectivo. Dado que los comportamientos de los avatares virtuales y de los robots se expresan habitualmente mediante lenguajes de marcado, se ha realizado un estudio en profundidad de los diferentes lenguajes de marcado existentes tanto para avatares virtuales, así como para robots afectivos.

Finalmente se ha realizado también un análisis del estado del arte de los robots en la educación, particularizando en el uso de NAO en este contexto.

2.1.Lenguajes de marcas

Los lenguajes de marcas son lenguajes que sirven, entre otras cosas, para definir la estructura de los documentos con el fin de intercambiar información entre diferentes sistemas, sin especificar cómo debe ser tratada la información que contienen.

En esta memoria se propone la implementación de un lenguaje de marcas para el intercambio de información sobre las actuaciones del robot, por lo que tal lenguaje ha de permitir la especificación de movimientos, gestos y diálogos (con tonalidad y volumen de voz).

Como se ha comentado anteriormente, no sólo existen lenguajes de marcas para robots, sino también avatares virtuales, y éstos podrían ser de también de gran interés para

nuestro propósito. Por tanto, en los apartados siguientes se describirán los lenguajes de marcas de mayor relevancia tanto para robots, como para avatares virtuales, que incluyen la representación de estados afectivos.

2.1.1. RoboML

RoboML es un lenguaje de marcado basado en XML (eXtensible Markup Language) para la programación de robots. Este lenguaje de marcas, tal como se describe en [Choi et al., 2014] [Shelton et al., 2012] provee métodos para el control del robot a través de internet mediante agentes embebidos, agentes de interfaz y agentes de proxy.

RoboML se basa en ontologías, y tal como se presenta en [Makatchev Maxim et al., 2000], implementa elementos de la ontología del hardware como son: “robot”, “wheel”, “motor”, “sensor”, “position” o “controller”. Este lenguaje además permite la incorporación de nuevas ontologías, que lo dota de una gran flexibilidad.

2.1.2. AIML

AIML, al igual que RoboML es un lenguaje de marcado basado en XML, pero a diferencia de RoboML, fue diseñado para el primer chatbot artificial online: A.L.I.C.E (Artificial Linguistic Internet Computer Entity Chatterbot). Las principales aplicaciones de los sistemas de chatbot se dividen en dos categorías: entretenimiento y acceso a la información [Liu et al., 2015], y se aplican cada vez más en los sistemas inteligentes educativos, habiéndose probado en todas las fases de los procesos educativos [Goc et al., 2007].

En [Claudia et al., 2007] se describe a AIML como un lenguaje en el que se establecen categorías de las posibles preguntas y expresiones del usuario, así como de las correspondientes respuestas.

En [Ahmed, 2005] se identifican las etiquetas más importantes de un documento AIML, tales como: “topic name”, “category”, “pattern” o “template”.

Además, es posible la implementación de AIML junto con ontologías que definen el modelo de diálogo (mejorando así el sistema de diálogo), tal como se demuestra en [Paper et al., 2014].

AIML se ha utilizado en varios estudios relevantes para este proyecto, como son:

- [Ahmed, 2005], donde se propone una arquitectura para desarrollar un bot de análisis del habla y sintonización de voz.
- [Claudia et al., 2007], donde se plantea el desarrollo de un Agente Inteligente Conversacional que hace las veces de un experto humano en Orientación Vocacional.
- [Goc et al., 2007], donde se propone el sistema LISE, un chatbot que pretende enseñar al estudiante ciertos conceptos asociados al material del curso.
- [Núñez et al., 2014], donde se implementa AIML en un robot, dotándolo de la capacidad de interactuar de forma natural con varias personas a la vez, y con la posibilidad de asistir a profesores dentro del aula.

Este lenguaje de marcado ha sido implementado además en aplicaciones móviles, como es el caso de CallMom [Riccardi, 2014] y en avatares 3D como en [Alencar et al., 2014] [Almajano and Lopez-sanchez, 2013].

2.1.3.FML y BML

FML (Functional Markup Language) y BML (Behavior Markup Language) son dos lenguajes de marcado complementarios. Ambos están basados en XML. FML ha sido diseñado para representar la estrategia de un agente: sus intenciones, objetivos y planes [Heylen et al., 2008]. Describe qué tiene que realizar el agente (robot, avatar,...), sin especificar cómo lo tiene que realizar [Ribeiro et al., 2013]. Recíprocamente, BML sirve para especificar cómo un agente tiene que realizar acciones previamente descritas en FML [Kopp et al., 2006].

Tanto FML como BML son lenguajes independientes de la plataforma física (robot) o virtual (avatar), como se presenta en [Paiva et al., 2012]. Por tanto, estos lenguajes de marcado permiten dotar de capacidad de expresión oral y/o gestual tanto a un robot como a un avatar virtual.

Existe un framework bastante utilizado llamado SAIBA (Situation, Agent, Intention, Behavior, Animation) que hace uso de ambos lenguajes de marcado tanto para especificar la intencionalidad de un determinado agente, como su comportamiento.

FML y BML han sido implementados en numerosos estudios en la literatura y para diferentes plataformas:

- En [Ribeiro et al., 2013] se presenta un proyecto en el que se desarrolla un robot tutor empático que juega a un juego colaborativo en una pantalla multitáctil.
- En [Paiva et al., 2012] se describe un robot cuentacuentos. El comportamiento del robot se realiza a través de Greta: una plataforma de tiempo real para el control multimodal de agentes conversacionales (robots, avatares virtuales, ...) que sigue el flujo marcado por SAIBA.

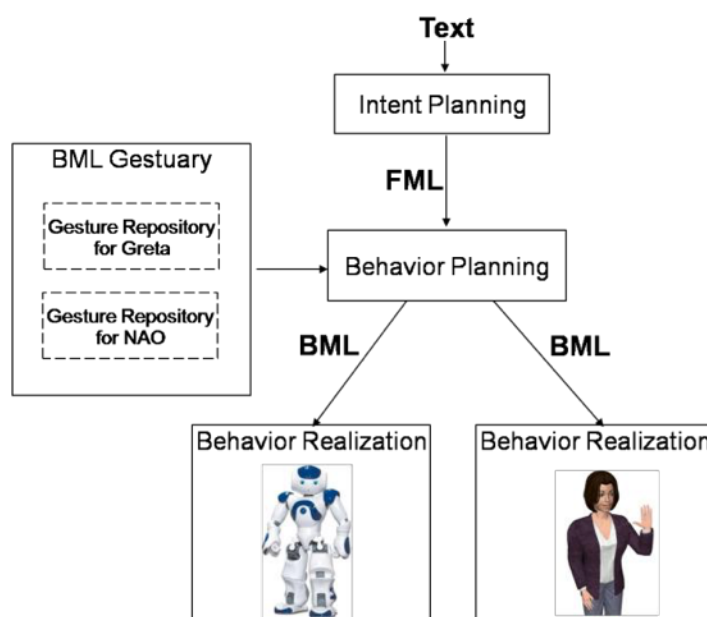


Figura 2.1: La plataforma Greta controlando los comportamientos de NAO (izquierda) y un agente virtual (derecha). (fuente: [Paiva et al., 2012])

- En [Lohse et al., 2012] se desarrolla un sistema para la transferencia de comportamientos a diferentes tipos de agente (robots y agentes virtuales). En este estudio se implementa un prototipo del sistema propuesto para un agente virtual 3D, para el robot FLOBI y para el robot NAO.

2.1.4. VHML

VHML (Virtual Human Markup Language) es un lenguaje de marcado basado en XML diseñado para gestionar los distintos aspectos de los avatares virtuales, incluyendo elementos de animación facial, síntesis de voz, animación corporal y gestión de interacción con diálogos y representación emocional [Beard et al., 2001][Prendinger et al., 2011]. Permite descripciones de alto nivel, sin detallar cómo tienen que actuar los avatares [Jovanova et al., 2009]. El lenguaje ofrece la posibilidad de derivar los estímulos y respuestas a otros dispositivos diferentes de un avatar virtual mediante la etiqueta <target>, además de mantener una interacción multimodal con los usuarios.

Se compone de la siguiente serie de sublenguajes de marcado:

- EML (Emotion Markup Language)
- GML (Gesture Markup Language)
- SML (Speech Markup Language)
- FAML (Facial Animation Markup Language)
- BAML (Body Animation Markup Language)
- XHTML (eXtensible HyperText Markup Language)
- DMML (Dialogue Manager Markup Language): este lenguaje de marcado está basado en AIML.

Su estructura, de forma esquemática se muestra en la figura 2.2.

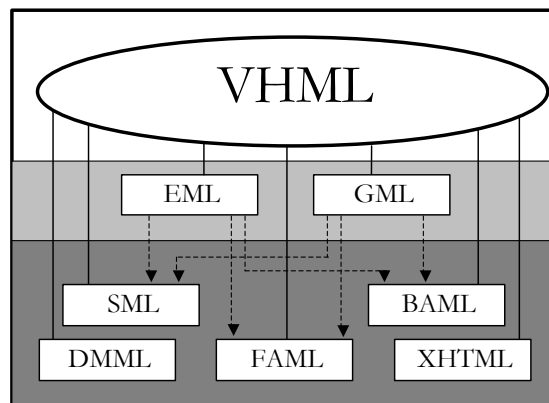


Figura 2.2: Estructura de VHML. (fuente: [Beard et al., 2001]).

VHML se basa en DTD (Document Type Definition) para la descripción de las reglas de la estructura de los lenguajes.

Existen estudios interesantes para este proyecto, donde se implementa el lenguaje VHML para el control de distintos dispositivos:

- En [Pandzic and Forchheimer, 2003][Gustavsson et al., 2002][Marriott and Shortland-Jones, 2003] se crea una cabeza virtual en tres dimensiones y a través de VHML se la

dota de expresión emocional, así como de interacción con el usuario a través del diálogo.

- En [Marriott et al., 2006] se desarrolla un sistema para asistir a personas mayores, basado en el sistema “Mentor”, que interacciona con el usuario a través de diálogos textuales.

VHML por lo tanto dota a un personaje virtual de las características necesarias para interactuar de forma natural con personas.

En [Marriott, 2011] se concluye que se trata de un lenguaje de marcado innovador y se destaca la importancia de dirigir la forma en la que un humano virtual interactúa con usuarios de manera que aparente ser más humano y creíble.

2.1.5.MPML

MPML es un lenguaje de marcado basado en XML, diseñado y desarrollado para facilitar las presentaciones multimodales por parte de agentes en entornos virtuales [Tsutsui et al., 2000]. Tiene las siguientes características:

- Independencia de la plataforma
- Simplicidad
- Sincronización de medios
- Control del Agente
- Control de la presentación interactiva

El lenguaje posee un módulo de conversación que se apoya en el sistema AIML [Mori et al., 2003]. En la Figura 2.3 se muestra un diagrama que resume esta interacción.

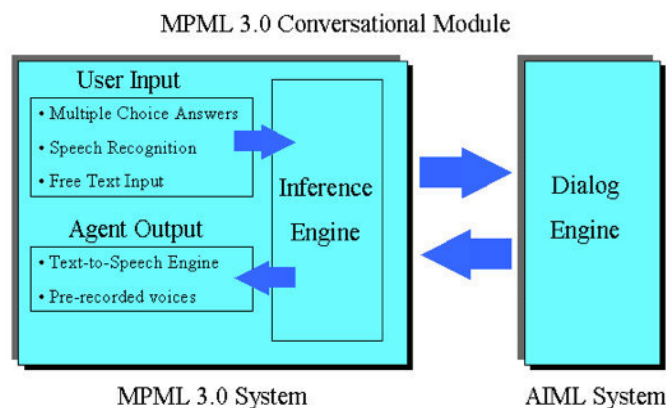


Figura 2.3: Módulo de conversación de MPML (fuente: [Mori et al., 2003]).

Además de dotar al agente virtual de la capacidad de realizar presentaciones y de la capacidad de diálogo, MPML también dota al personaje de la capacidad de expresar emociones a través de las acciones y el habla [Zong et al., 2000].

Mientras que otros lenguajes de marcado desarrollados para personajes virtuales presentan sofisticados mecanismos de control y están orientados a programadores o

diseñadores expertos, MPML está diseñado para usuarios no expertos, permitiendo dirigir los comportamientos de múltiples personajes animados [Prendinger et al., 2004].

Otro punto importante de este lenguaje de marcado orientado a las presentaciones, es la posibilidad de llevarlo a otras plataformas. No obstante, está diseñado para dos dimensiones, lo que hace complicado trasladarlo a un mundo tridimensional.

En este sentido, en [Prendinger et al., 2011] se describe el desarrollo de MPML3D, que adapta el lenguaje de marcado MPML a la realidad virtual tridimensional.

Este nuevo lenguaje implementa además percepciones internas y externas, así como gestos y movimientos en todos los sentidos (x,y,z). Estas nuevas características unidas a la posibilidad de llevar el lenguaje a cualquier plataforma, podrían habilitar a MPML en un mundo tridimensional real, en el que los agentes de presentación fueran robots.

Tanto MPML como MPML3D han sido objeto de numerosos estudios relevantes para este proyecto:

- En [Mori et al., 2003] se muestra una presentación interactiva donde se desarrollan conversaciones naturales entre el agente presentador y el usuario.
- En [Yang et al., 2004] se desarrolla un agente de presentación en Flash.
- En [Ito, 2005] se introduce un agente de presentación en una aplicación web ya existente.
- En [Saeyor et al., 2003] se desarrolla un agente de presentación para plataformas móviles.
- Finalmente en [Prendinger et al., 2011] se ha implementado MPML3D en mundos tridimensionales como Second Life y OpenSim dando como resultado una interacción eficaz entre los personajes del mundo virtual y facilitando el control verbal y no verbal de estos personajes.

2.1.6. EmotionML

EmotionML es un lenguaje de marcado basado en XML cuyo objetivo es representar las emociones humanas [Meftah, 2013]. Se puede implementar para el reconocimiento automático de estados emocionales de un usuario y para la generación de emociones [Schröder et al., 2011][Cruz-Lara et al., 2014].

En este lenguaje, las emociones pueden estar representadas con cuatro marcas diferentes:

- <category>
- <dimension>
- <appraisal>
- <action-tendency>

EmotionML abarca aspectos temporales, permitiendo establecer el principio y el final de un estado emocional [Meftah, 2013].

EmotionML ha sido implementado en varios estudios relevantes para este proyecto:

- En [Schuller et al., 2014] se desarrolla una aplicación web para la inclusión de niños con autismo donde se detecta el estado emocional de los niños mediante un analizador de voz.
- En el proyecto ITEA2 METAVERSE1 se realizan análisis de sentimientos y reconocimiento de emociones derivados del análisis sintáctico y semántico de frases de herramientas de chat o de redes sociales tales como Twitter [Cruz-Lara et al., 2014].

A pesar de que EmotionML facilita el desarrollo de un sistema interoperable, estable y durable, y que comprende la representación y el intercambio de los estados emocionales, no abarca la gestión de las emociones, ni propone reglas para esta gestión [Meftah, 2013] [Cruz-Lara et al., 2014].

2.2.Reconocimiento de estados emocionales a partir de análisis de logs

La investigación en computación afectiva ha ido creciendo paulatinamente en las últimas décadas. Un sistema inteligente capaz de reconocer afectos (estados afectivos o emociones) resulta más efectivo y natural [Khan et al., 2013].

Las medidas afectivas se pueden agrupar en tres áreas diferentes: fisiológicas, de comportamiento y psicológicas [Salmeron-majadas, 2014]. Los métodos convencionales de identificación emocional (análisis de expresión facial, análisis de entonación de voz, etc.) utilizan herramientas intrusivas, caras e inviables para situaciones reales [Khan et al., 2013].

Puesto que el propósito de este proyecto es precisamente facilitar el desarrollo de sistemas realistas, no intrusivos y de bajo coste que puedan utilizarse en diversos contextos educativos, la identificación de emociones se basará en las interacciones del usuario con el teclado y el ratón. Este tipo de entrada de información (teclado y ratón) es la menos intrusiva y no requiere ningún dispositivo extraordinario [Rajput and Vijayavargiya, 2015].

Existen multitud de estudios sobre el análisis de la interacción a través de teclado y ratón que no han sido, sin embargo, particularizados para el caso de alumnos de primaria, alumnos que interactúan de forma peculiar (en particular, realizan más movimientos bruscos y aleatorios).

Las fases de un proceso de reconocimiento de emociones basado en análisis de logs son, tal como se muestra en la literatura: recogida de datos, etiquetado de estados emocionales, y análisis de las dinámicas de tecleo e interacciones de ratón.

Cada una de estas fases será descrita en los siguientes apartados.

2.2.1.Recogida de datos

En esta primera fase se obtiene la información de muestra. Su objetivo es registrar los diferentes estados emocionales de los alumnos junto a los correspondientes datos de interacción de teclado y de ratón.

La recogida de datos implica por tanto dos tareas:

- Recopilación de los datos de interacción de teclado y de ratón: En la literatura existen estudios donde la recopilación de datos de interacción se realiza mediante un software propio e independiente de cualquier software que se está ejecutando en el sistema operativo, como se muestra en [Zimmermann et al., 2003][Rajput and Vijayavargiya, 2015][Khanna and Sasikumar, 2010].
- Recopilación de material auxiliar para la identificación de los estados emocionales por parte de expertos. Así por ejemplo, existen estudios donde se realiza una recogida de material audiovisual para ayudar a la identificación de los estados emocionales. Es el caso de [Salmeron-majadas, 2014][Salmeron-majadas et al., 2014], donde se registra en formato video el escritorio del usuario.

2.2.2. Etiquetado de estados emocionales

Los datos de las interacciones de teclado y ratón, así como los videos registrados, no indican de forma explícita los estados emocionales de los alumnos, por lo que para analizar las correlaciones entre patrones de interacción y estados emocionales es preciso etiquetar previamente los datos de interacción recopilados.

En la literatura existen dos métodos muy utilizados a la hora de identificar el estado emocional de los alumnos:

- La encuesta SAM (Self-Assessment Manikin): Se trata de una evaluación ilustrada para identificar y medir estados emocionales [Bradley and Lang, 1994]. Este método ha sido implementado en [Salmeron-majadas et al., 2014] [Khanna and Sasikumar, 2010] [Khan et al., 2013].
- La visualización de videos por parte de expertos, tal y como se describe en [Salmeron-majadas, 2014] [Salmeron-majadas et al., 2014].

2.2.3. Análisis de las interacciones

Una vez obtenidas las interacciones de teclado y ratón, se ha de analizar la muestra con el objetivo de determinar la correlación entre unos ciertos parámetros que caractericen las interacciones y el estado emocional del alumno.

En un gran número de aproximaciones propuestas en la literatura sobre detección de emociones se considera una única fuente de datos (ratón o teclado) como indicador afectivo. Sin embargo en [Khanna and Sasikumar, 2010] se demuestra que la detección de estados afectivos mejora notablemente cuando se combina el análisis de las interacciones del teclado con el de las interacciones de ratón.

En este apartado se describen los métodos existentes para la identificación de estados emocionales mediante el análisis de las interacciones de teclado y de las interacciones de ratón.

A. Análisis de las interacciones de teclado

El análisis de las interacciones de teclado tiene por objetivo la identificación de los estados emocionales del alumno a través del teclado, dando importancia no tanto a lo que se teclea como a la forma en que se teclea [Khan et al., 2013].

En las interacciones de teclado pueden considerarse multitud de parámetros que han sido estudiados en la literatura con el objetivo de identificar estados emocionales.

Algunos parámetros interesantes para este proyecto han sido analizados en diferentes estudios. Algunos estudios interesantes son:

- [Rajput and Vijayavargiya, 2015], donde se analizan los siguientes parámetros:
 - Latencia de tecleo (intervalo entre la suelta de una tecla y la presión de la siguiente tecla).
 - Tiempo de presión (tiempo en el que una tecla se mantiene presionada).
 - Secuencia (lista de teclas pulsadas consecutivamente).
 - Velocidad de tecleo (número total de teclas pulsadas por minuto)
- [Khanna and Sasikumar, 2010], donde se analizan los siguientes parámetros:
 - Velocidad de tecleo.
 - Número de caracteres tecleados en 5 segundos.
 - Tiempo total para el tecleo.
 - Número total de retornos utilizados.
 - Tiempo de paradas.
- [Khan et al., 2013], donde además, se tiene en cuenta si las teclas presionadas son mayúsculas, minúsculas, numéricas o especiales.

Además de diferentes parámetros de interacción de teclado, en la literatura se documentan diferentes aproximaciones para recopilar estos parámetros, así como el estado emocional en el que se encuentra el alumno. Algunos estudios interesantes son:

- [Rajput and Vijayavargiya, 2015], donde los participantes escriben un texto aleatorio en un estado mental conocido, inducido por manipulación experimental.
- [Khan et al., 2013], donde los participantes escriben un texto fijo (concretamente fragmentos de la obra "Alicia en el país de las maravillas").

Como se puede observar, estos estudios están diseñados especialmente para la inserción de texto libre o predefinido en un tiempo ilimitado. Sin embargo, en este proyecto el uso del teclado va a estar bastante limitado, lo que hace que ciertos parámetros tales como el tiempo total para el tecleo o el tiempo de paradas no cobren especial sentido.

Por lo tanto, en este proyecto se analizarán los siguientes parámetros de tecleo:

- Número total de retornos.
- Número total de teclas suprimir.
- Latencia de tecleo (intervalo entre la suelta de una tecla y la presión de la siguiente tecla).
- Tiempo de presión.

B. Análisis de las interacciones del ratón

El análisis de las interacciones de ratón tiene por objetivo la identificación de los estados emocionales del alumno en función de su manejo del ratón.

En [Khanna and Sasikumar, 2010] se demuestra que el análisis combinado de las interacciones de teclado y las interacciones de ratón mejora notablemente los resultados obtenidos.

Existen numerosos parámetros de ratón analizados en la literatura. Algunos estudios interesantes son:

- [Zimmermann et al., 2003], donde se analizan los siguientes parámetros:
 - Número de clicks de ratón por minuto.
 - Distancia media en pantalla de los clicks de ratón (desde que se presiona hasta que se suelta).
 - Distancia total de los movimientos de ratón en píxeles.
 - Distancia media de un movimiento de ratón.
 - Número y tamaño de las pausas en el movimiento de ratón.
 - Número de eventos “bruscos” (más de 5 cambios de dirección en 2 segundos).
 - Máximo, mínimo y media de la velocidad de ratón.
- [Salmeron-majadas et al., 2014] , donde se analizan:
 - La distancia euclídea entre las posiciones de los clicks de ratón, distancia global y distancia global cubierta entre dos presiones de botón.
 - La diferencia entre la distancia cubierta y la distancia euclídea calculada.
 - Las duraciones de tiempo entre las combinaciones de eventos.

En este proyecto se analizarán los siguientes parámetros de ratón :

- Número de clicks de ratón por minuto,
- Distancia media en pantalla de los clicks de ratón.
- Distancia total de los movimientos de ratón en píxeles.
- Número y duración de las pausas en el movimiento de ratón.
- Número de eventos “bruscos”.
- Máximo, mínimo y media de la velocidad del ratón.

2.2.4.Estados emocionales

En los seres humanos existe una serie de posibles estados emocionales más o menos complejos que son innatos e iguales para todos. Las emociones más complejas se pueden definir como una combinación de las emociones más básicas. Se puede decir que existen entre dos y veinte “emociones básicas” tales como felicidad, miedo, amor, sorpresa, tristeza, etc. [Khan et al., 2013].

En la literatura se han estudiado diferentes estados emocionales de alumnos, según su interacción con el teclado y el ratón. Algunos estudios interesantes son:

- [Rajput and Vijayavargiya, 2015], donde se han estudiado los siguientes estados emocionales: confianza, tristeza, nerviosismo, felicidad, cansancio e indecisión.

- [Khanna and Sasikumar, 2010], donde se han estudiado los siguientes estados emocionales: estado neutro, desprecio, enfado, irritación, inquietud, aburrimiento, tristeza, disgusto, relajación, alivio, satisfacción, deleite, excitación, felicidad y satisfacción.
- [Bradley and Lang, 1994], donde se han estudiado los siguientes estados emocionales: felicidad, alegría, satisfacción, ilusión, relajación, tristeza, irritación, insatisfacción, melancolía, desesperación y aburrimiento.

En este proyecto, tras llevar a cabo la experiencia de usuario que relatamos en el capítulo 1.4. Objetivos y Metodología y mantener entrevistas con profesores de primaria que realizan actividades con ordenadores, hemos concluido que en el caso de los alumnos de este periodo educativo existen tres estados principales que condicionan significativamente la naturaleza de la intervención pedagógica. Estos estados no se corresponden con emociones específicas sino con categorías de emociones, y sería más apropiado designarlos estados cognitivo-afectivos:

- **Concentrado:** El alumno está realizando los ejercicios y tiene una interacción normal con el programa educativo. Implica emociones de asombro, curiosidad, confianza, entusiasmo, interés, intriga ...
- **Distraído:** El alumno está distraído realizando otras tareas diferentes a las asignadas (utilizando otros programas), o bien la interacción con el programa educativo es irregular. Implica emociones de aburrimiento, desgana, indiferencia, pereza, ...
- **Inactivo:** El alumno tiene dudas, está hablando con los compañeros y no existe interacción con el teclado y el ratón, o es una interacción mínima. Implica emociones de aburrimiento, confusión, desánimo, desconcierto, enfado, frustración, impotencia, irritación, pesimismo, ...

2.2.5. Métodos de aprendizaje

Una vez etiquetadas las interacciones de teclado y ratón se entrena un modelo de minería de datos para la identificación automática de estados emocionales.

En la práctica, los modelos de minería de datos pueden ser de dos tipos: predictivos y descriptivos. Los modelos predictivos pretenden estimar valores futuros o desconocidos de variables de interés. Los modelos descriptivos, en cambio, identifican patrones que explican o resumen los datos [Hernández Orallo et al., 2004].

De cara a la generación de un modelo para la identificación de estados emocionales, en la literatura se han propuesto aproximaciones que utilizan tanto métodos predictivos como métodos descriptivos.

En [Salmeron-majadas et al., 2014], se han utilizado los siguientes algoritmos:

- C4.5
- Naïve Bayes
- Bagging
- Random Forest
- AdaBoost

Sin embargo, en [Rajput and Vijayavargiya, 2015], los métodos son diferentes:

- Análisis discriminatorio

- Análisis bayesiano
- k-vecinos
- redes neuronales artificiales
- árboles de decisión

Como se puede observar, tanto en [Salmeron-majadas et al., 2014] como en [Rajput and Vijayavargiya, 2015] se han utilizado tanto métodos predictivos (clasificación, regresión, categorización,...) como métodos descriptivos (agrupamiento, correlaciones, reglas de asociación,...).

En nuestro caso se han implementado, al igual que en las referencias señaladas, métodos predictivos y métodos descriptivos.

Tanto en [Rajput and Vijayavargiya, 2015] como en [Cocea and Weibelzahl, 2009] se utiliza el método k-vecinos. Aunque en [Rajput and Vijayavargiya, 2015] no se indican los resultados obtenidos, en [Cocea and Weibelzahl, 2009] este método ofrece muy buenos resultados en el análisis de logs, por lo que será uno de los métodos que se implementen.

Los árboles de decisión (C4.5) también han dado buen resultado en [Salmeron-majadas et al., 2014].

Tal como se ha indicado, en [Salmeron-majadas et al., 2014] [Rajput and Vijayavargiya, 2015] se han implementado las redes neuronales artificiales y Naïve Bayes con buenos resultados, por lo que resulta interesante comprobar su eficacia en nuestro estudio.

Un método que, a juzgar por la literatura, no ha sido implementado en este ámbito de aplicación, y cuya eficiencia podría resultar interesante comprobar es el de las máquinas de vectores soporte. Este método de clasificación lineal está indicado para problemas de alta dimensionalidad, como es el caso en este proyecto, en el que se dispondrá de 10 dimensiones. Hemos juzgado de interés investigar el comportamiento de dicho método de clasificación aplicado a la identificación de estados emocionales a través del análisis de interacciones de teclado y ratón.

En virtud de lo anteriormente expuesto, resumiendo, en este proyecto hemos realizado un análisis de los datos mediante los siguientes métodos:

- K-vecinos
- C4.5
- SVM
- Redes neuronales artificiales
- Naïve Bayes

2.3.Métodos de elicitación de conocimiento para el desarrollo de software educativo afectivo

Con el fin de analizar los estados afectivos de los alumnos, además de recopilar los datos de las interacciones de teclado y ratón y las grabaciones tanto del escritorio como de la webcam, es conveniente también utilizar otros métodos de observación para la elicitación de conocimiento durante las sesiones. Estos métodos proporcionan información adicional para el etiquetado de estados afectivos que ha de sincronizarse con los datos de los logs obtenidos por el software educativo que los estudiantes están utilizando en el momento del estudio [Ocumpaugh, 2012].

En este proyecto se pueden distinguir dos tipos de conocimiento que debe ser incorporado:

- Un conocimiento que permita el reconocimiento de la respuesta afectiva de los alumnos.
- Un conocimiento que permita identificar la respuesta pedagógica afectiva oportuna en cada circunstancia.

2.3.1. Tipos de observadores

Los métodos de observación pueden utilizarse en numerosos contextos diferentes, y por tanto contemplan la posibilidad de diferentes tipos de observadores. En nuestro caso estos observadores pueden ser los mismos alumnos que anoten las emociones experimentadas de primera mano mediante auto informes (como puede ser el caso de SAM), observadores externos que pertenezcan al entorno educativo (profesores o alumnos) u observadores externos que no pertenezcan a dicho entorno [Porayska-Pomsta et al., 2013].

En nuestro caso, si los profesores realizasen la función tanto de observador como de tutor, muy probablemente se les pasarían por alto anotaciones relevantes mientras estuviesen atendiendo a algún alumno.

Por tanto, hemos utilizado observadores-anotadores externos, colocados en sitios estratégicos de la sala de forma que no interfirieran en la experiencia y liberasen al tutor de la tarea de observación-anotación.

2.3.2. Métodos de elicitación

En este proyecto se han aplicado dos métodos de elicitación diferentes: uno que permite reconocer la respuesta afectiva de los alumnos, y otro que permite identificar la respuesta pedagógica afectiva oportuna en cada circunstancia.

A. Métodos para el reconocimiento de la respuesta afectiva de los alumnos

El objetivo es identificar el estado emocional de los alumnos asociado a diferentes circunstancias de aprendizaje. Para ello se realizan anotaciones de dichos estados emocionales durante las sesiones de observación.

Las anotaciones de las observaciones se pueden realizar de forma concurrente o de forma retrospectiva.

En el caso de los anotadores externos, la forma concurrente consiste en que varias personas observan cómo los estudiantes realizan las tareas y describen en tiempo real los estados afectivos de los alumnos [Ocumpaugh and College, 2014].

Para los anotadores externos existe también la posibilidad de realizar las observaciones de forma retrospectiva, mediante la visualización de videos y la correspondiente anotación de estados afectivos en diferido [Ocumpaugh and College, 2014].

En esta experiencia se han realizado grabaciones generales de las sesiones para su posterior análisis. Sin embargo, hemos juzgado interesante realizar asimismo anotaciones de forma concurrente durante las sesiones, ya que en las grabaciones se pierden

perspectivas de las escenas grabadas, así como diferentes matices de la atmósfera afectiva que solo pueden percibirse o intuirse en vivo.

Para ello se ha seguido el método BROMP (Baker-Rodrigo Ocumpaugh Monitoring Protocol) descrito en [Porayska-Pomsta et al., 2013]. Este método se ha concebido para estudiar los comportamientos y afectos en un entorno basado en aprendizaje por ordenador.

Un estudio particularmente relevante para este proyecto es [Ocumpaugh and College, 2014] donde se comenta el uso de una herramienta llamada HART (Human Affect Recording tool), aplicación Android diseñada para realizar anotaciones afectivas de distintos alumnos siguiendo el método BROMP.

B. Métodos para la identificación de la respuesta pedagógica afectiva

El objetivo es determinar la respuesta pedagógica afectiva oportuna según el estado emocional del alumno y las circunstancias del aprendizaje. Un estudio de especial relevancia para este proyecto, donde se implementan metodologías para la identificación de respuestas pedagógicas, es [Blazar, 2013], cuyo propósito es diseñar un nuevo método para evaluar al profesorado. Para ello se implementan dos métodos de evaluación: CLASS (Classroom Assessment Scoring System) y MQI (Mathematical Quality of Instruction). Para aplicar estos métodos se realizan grabaciones de las clases que posteriormente se analizan.

Al igual que en [Blazar, 2013], en este proyecto se han realizado grabaciones de vídeo para posteriormente analizar las respuestas pedagógicas.

2.4.Scratch en la educación

La programación se considera una competencia importante a la hora de resolver problemas lógicos [Kalelioğlu and Gülbahar, 2014]. Adicionalmente, en la actualidad los estudiantes perciben que aprenden más utilizando un ordenador que escribiendo a mano o recibiendo enseñanzas sentados delante del televisor [Umrani-khan, 2000].

Scratch es un lenguaje de programación concebido para que los jóvenes comprendan mejor el concepto de programación, permitiendo crear historias interactivas, animaciones, juegos, música y arte a través de paletas de bloques organizadas por secciones [Valle and Salgado, 2013]. Los niños deben desarrollar un pensamiento algorítmico, creatividad y destreza para resolver problemas durante su ciclo escolar, y Scratch resulta una herramienta muy útil a la hora de ejercitar dichas destrezas.

Scratch se ha integrado en el currículum de muchos países como México [Valle and Salgado, 2013], Reino Unido [Wilson and Moffat, 2010], Turquía [Kalelioğlu and Gülbahar, 2014] o Colombia [López García, 2012].

2.4.1.Scratch en educación primaria

Según sugieren la literatura especializada y las opiniones de los expertos, la programación puede mejorar las capacidades de resolver problemas de los estudiantes a cualquier edad. En lo que respecta a los alumnos de primaria, Scratch resulta una herramienta sencilla de

utilizar, tal y como se expone en el estudio [Kalelioğlu and Gülbahar, 2014], donde se muestra que la mayor parte de los estudiantes que participaron en la experiencia reportada, además de haber disfrutado de esta experiencia desea mejorar sus conocimientos de programación y es capaz de resolver un mismo problema de diferentes formas.

Scratch ha resultado la herramienta seleccionada para este proyecto debido a sus siguientes características:

- Es gratuito, de código abierto, en permanente evolución, y dispone de una comunidad en línea para usuarios del entorno y otra para educadores [López García, 2012].
- Es ampliamente utilizado en la educación, y concretamente en educación primaria [Valle and Salgado, 2013].
- Se ha mostrado pedagógicamente efectivo, de fácil uso y capaz de mantener la motivación de los niños [Valle and Salgado, 2013].
- Ayuda a desarrollar habilidades interesantes como imaginar, crear, jugar, compartir y reflexionar [López García, 2012], o competencias como el desarrollo del pensamiento computacional, adquisición de conocimiento conceptual académico y habilidades de planificación [López García, 2014].
- Como suele mantener una alta motivación en los niños, se puede presumir que los niños no se aburrirán en las sesiones, a menos que se encuentren perdidos, lo que simplifica el análisis de los estados emocionales en nuestra primera aproximación al problema.
- Resulta particularmente apropiado, ilustrativo y motivador para los niños que un robot tutorice su primera experiencia de aprendizaje de la programación. Como trabajo futuro, puede plantearse la posibilidad de extender Scratch con un módulo para el control del propio robot tutor con que los niños puedan también experimentar.

2.5. Robótica en educación

La enseñanza constituye una interacción social entre profesor y estudiantes [Saerbeck et al., 2010]. Tal como se expone en [Pellegrini and Blatchford, 2000], en educación primaria la interacción social entre compañeros y con el profesor es muy importante. Los compañeros proveen un apoyo social y emocional vital en cada nuevo curso, para hacer frente a las circunstancias estresantes. Las relaciones entre los compañeros afectan a la calidad de los procesos educativos y de aprendizaje, y pueden también causar dificultades que deben ser resueltas por los profesores.

En el ámbito educativo existen multitud de investigaciones que tienen por objetivo desarrollar agentes virtuales como recurso pedagógico, ya que pueden simular una interacción social real. Sin embargo, tal como se expone en [Saerbeck et al., 2010] la apariencia del agente educativo tiene un impacto significativo en el comportamiento del usuario. Se ha demostrado que la percepción de un compañero interactivo puede ser mejorada por un robot físico. No obstante, es preciso tener en cuenta que la aceptación del robot por parte de los profesores es vital para asegurar su correcto uso como herramienta asistencial [Fridin and Belokopytov, 2013].

La integración de la robótica en nuestra vida cotidiana ha ido creciendo paulatinamente, tanto en el hogar como en la escuela [Mubin et al., 2013]. Los robots se han introducido en varios campos para asistir a los humanos en diferentes situaciones. Los medios de aprendizaje electrónicos suponen una revolución dentro del sistema educativo, proveyendo una forma eficiente de impartir amplios conocimientos [Veena Vijayan, 2014]. Los robots educativos influyen de forma positiva contribuyendo a que los estudiantes obtengan mejores resultados en los exámenes [Catlin and Robertson, 2012] y consiguiendo que aumente su interés más que los libros u otros recursos audiovisuales [Mubin et al., 2013].

Dentro de la educación, inicialmente la robótica fue enfocada desde el prisma del constructivismo, donde los alumnos aprenden a resolver problemas a partir de la construcción de un artefacto físico. Más tarde se introdujo la noción de constructivismo social propuesta por Vygotsky, que aplica la mayoría de las metodologías basadas en tutores robots en educación [Mubin et al, 2013].

Los robots socialmente interactivos idealmente deberían tener una serie de características tales como: expresar o percibir emociones, comunicar con alto nivel de diálogo, aprender/reconocer modelos de otros agentes, establecer/mantener relaciones sociales, utilizar señales naturales, exhibir personalidad distintiva y aprender/desarrollar competencias sociales [Fong et al., 2002].

Debido a que los resultados académicos han mejorado con la introducción de robots sociales, y a que el impacto de estos robots es mayor que el de los agentes virtuales consiguiendo una mayor motivación en los alumnos, este proyecto se planteó también la implementación de un sistema de traducción de agentes virtuales a robots capaz de dotar a un robot de las características sociales de los agentes virtuales.

2.5.1. Robots socialmente interactivos

Las tecnologías de asistencia social basadas en robótica tienen un gran potencial para conseguir herramientas educativas eficientes [Keren et al., 2012].

En los últimos años ha habido un incremento en el desarrollo de robots socialmente interactivos con la habilidad de interpretar características sociales [Vouloutsi et al., 2014] que los habilitan para interactuar de forma natural con los humanos [Salam and Chetouani, 2015].

En [Fong et al., 2002], en base a un extenso estudio del estado de la investigación en robots socialmente interactivos se concluye que para facilitar la credibilidad en la interacción entre el robot y la persona el robot tiene que incorporar emociones artificiales, llamando la atención sobre las siguientes cuestiones:

- El habla es un método altamente efectivo de comunicar emociones. Hasta el momento se ha utilizado en algunos sistemas robotizados para expresarlas.
- El comportamiento expresivo de las caras de los robots, generalmente no es realista. En lugar de usar actuadores mecánicos, existe otra aproximación que hace uso de técnicas de animación.
- El lenguaje corporal del robot indica cuál es su foco de atención.

- El diálogo es el proceso conjunto de comunicación. Este proceso entre el hombre y el robot puede suceder de varias formas: a bajo nivel, en forma de lenguaje no verbal y en forma de lenguaje natural hablado.
- La percepción humana permite a los robots sociales percibir el mundo como es. Para ello tienen que realizar seguimientos de las diferentes partes del cuerpo humano (cara, cuerpo, manos), e interpretar el habla humana, así como las expresiones faciales, los gestos y la actividad humana.

En la actualidad, la robótica asistencial, social y de servicio es un área de investigación donde la interacción humano-robot es determinante [Núñez et al., 2014]. Existen robots asistenciales y sociales destinados a la educación de niños preescolares como es el caso de Nao, utilizado en el proyecto Kindergarten Assistive Robotics (KAR) [Keren et al., 2012].

En el contexto de la asistencia médica existen experiencias de que el uso de robots para la asistencia a personas mayores mejora la calidad de vida de estas personas a las que, además, los robots hacen compañía [Hammer et al., 2015].

2.5.2. Robots educativos afectivos

El uso de robots para apoyar la enseñanza se ha vuelto un tema de investigación popular en los últimos años, en que se han demostrado útiles en la enseñanza de asignaturas tales como matemáticas o física [Capponi et al., 2010]. Estos robots, capacitados para identificar emociones a través del reconocimiento facial y gestual, pueden ofrecer una asistencia efectiva al profesor [Veena Vijayan, 2014].

En la educación especial existen estudios que demuestran la eficacia del uso de robots. En [Ismail et al., 2012] se plantea una experiencia donde se sustituye un tutor humano por un robot NAO en varias sesiones de terapia con niños con autismo (en inglés ASD: Autism Spectrum Disorder) con el objetivo de medir el tiempo de contacto visual entre el niño y el robot, y compararlo con el tiempo de contacto visual entre el niño y el tutor humano. En este estudio se concluye que la interacción con el robot humanoide NAO es beneficiosa, ya que mejora el tiempo de contacto visual.

Otra aproximación que resulta interesante en la educación especial es la que se muestra en [Robins et al., 2004], donde se pretende mejorar la interacción con extraños en niños con autismo. Este estudio se ha realizado con niños de entre 5 y 10 años de edad, y las pruebas han transcurrido en habitaciones utilizadas frecuentemente por estos niños. En [Robins et al., 2004] se establecen dos escenarios de interacción diferentes: un escenario en el que se cuenta con una persona desconocida para los niños, y otro escenario con un robot. Aquí se demuestra que el uso de robots mejora también la respuesta de los niños ante un desconocido en comparación con la interacción con personas, lo que permite que los niños con este desorden mejoren su capacidad de comunicación y de atención.

Además de utilizarse robots en la enseñanza especial, también se han utilizado para estudios con niños de primaria y secundaria.

En educación primaria existe ya un amplio número de estudios realizados con diferentes robots. Dentro de la enseñanza existen dos enfoques diferentes: el enfoque del robot tutor en que el robot juega el rol de maestro, enseñando las lecciones al alumno; y el enfoque en que el rol del robot consiste en recibir cuidados [Tanaka and Matsuzoe, 2012].

En [Tanaka and Matsuzoe, 2012] se plantea el enfoque del robot que recibe las lecciones de los alumnos, mejorando de forma notable su aprendizaje. Esto es debido al aumento de la motivación que suscita ver los progresos del robot durante el proceso de aprendizaje.

Al igual que con los niños de primaria, existen estudios que demuestran que los robots pueden ser buenos tutores asistiendo en clases de niños de entre 3 y 4 años, mejorando su pensamiento geométrico [Keren et al., 2012].

Cabe destacar que, en todos los estudios analizados hasta el momento, la presencia de un robot en el aula mejora notablemente la curva de aprendizaje de los alumnos. Sin embargo, a pesar de su motivación inicial ante los robots tutores, poco a poco los alumnos van perdiendo el interés en estos robots [Jimenez et al., 2015]. Es a raíz de estos resultados que se comienzan a realizar estudios sobre educación afectiva en el campo de la robótica, dotando a los robots de la capacidad de interpretar y expresar sentimientos.

Concluimos por tanto que existen suficientes estudios que avalan la hipótesis de que la utilización de robots, y particularmente de robots afectivos, en la educación puede mejorar los procesos de aprendizaje, tal como se concluye en [Saerbeck et al., 2010] [Capponi et al., 2010] [Jimenez et al., 2015] [Keren et al., 2012].

2.5.3. NAO en la educación

NAO es un robot humanoide desarrollado por la compañía francesa Aldebaran Robotics. Este robot mide 60 cm de alto, tiene 25 grados de libertad y está equipado con un acelerómetro, dos cámaras y otros muchos sensores entre los que se incluye un sensor táctil.

En [Keren et al., 2012] se presenta un estudio en el que el robot NAO ayuda a los profesores de preescolar realizando juegos educativos con los niños y monitorizando su desarrollo cuando realizan tareas específicas y en diferentes situaciones de aprendizaje.

En [Ribeiro et al., 2013] se presenta un robot tutor que asiste a los estudiantes en temas de geografía. Para ello utiliza el juego *Energities2* sobre una tablet de grandes dimensiones y el robot NAO. *Energities 2* es un juego basado en turnos para tres jugadores: dos humanos y un robot equipado con inteligencia artificial.

El objetivo de este estudio es el desarrollo de un robot tutor con la capacidad de percibir emociones experimentadas por los alumnos, de forma que mejoran sus estrategias pedagógicas. NAO actúa tanto de tutor como de jugador, colaborando con dos estudiantes en el desarrollo de una ciudad virtual enfocada a temas medioambientales. En este estudio se demuestra que el tutor NAO es capaz de reaccionar de forma creíble y empática ante los movimientos de los jugadores.

De igual forma, en [Tanaka and Matsuzoe, 2012] se realiza un estudio en un colegio de lengua inglesa para niños japoneses de entre 3 y 6 años de edad. Los autores de este estudio proponen un método para motivar en el aprendizaje de los verbos ingleses. Este enfoque es diferente en cuanto a robot tutor se refiere: el robot no enseña, sino que los niños ejercen de tutores del robot. En este estudio se ha utilizado a NAO como robot "alumno".

Además de utilizarse en la educación preescolar, NAO también se ha utilizado en investigaciones del ámbito de la educación especial. Tal y como se señaló en la sección anterior, en [Ismail et al., 2012] se propone un enfoque para mejorar el contacto visual de los niños con autismo utilizando un robot NAO en sesiones de terapia. Mediante

grabaciones, con la ayuda de expertos, se mide el tiempo de contacto visual y se compara con el tiempo de contacto visual cuando en la terapia interviene únicamente el tutor.

NAO ha sido objeto de estudio en diversas líneas de investigación. Es así como en [Paiva et al., 2012] se ha utilizado NAO para probar el framework SAIBA, o en [Salam and Chetouani, 2015] para probar un enfoque en que se mejora la interacción entre hombre y robot mediante el modelado del compromiso basado en el contexto.

NAO, por tanto, no sólo ha sido objeto de estudio en el contexto de la educación preescolar [Keren et al., 2012] y primaria [Ribeiro et al., 2013] [Tanaka and Matsuzoe, 2012], sino que también ha sido objeto de estudio en la educación especial [Ismail et al., 2012], y en todos los casos se han logrado buenos resultados en los objetivos marcados, consiguiendo incluso una gran aceptación por parte de adultos y niños [Fridin and Belokopytov, 2013].

En este proyecto se ha escogido a NAO como interfaz del sistema recomendador, debido por un lado a la buena aceptación que se refleja en los diferentes estudios analizados hasta el momento dentro de la educación primaria, y por otro lado a las grandes posibilidades funcionales y a la gran flexibilidad que ofrece el sistema incorporado en NAO (modificación del tono de voz, capacidad de escuchar, reconocimiento facial, amplio número de movimientos con todo el cuerpo, o expresión facial a base de luces de colores).

2.5.4. Robots y sistemas recomendadores

La robótica está evolucionando hacia robots completamente autónomos que trabajan en entornos desconocidos en simbiosis con la gente [Coradeschi and Saffiotti, 2006]. Para llegar a este punto, es necesario que los robots sean capaces de conocer el entorno que los rodea.

El proceso de adquisición de conocimientos es un componente clave para muchos robots sociales, permitiéndoles mejorar sus acciones en un entorno humano dinámico. Por esta razón, una de las aproximaciones a la implementación de sistemas de conocimiento en robots consiste en imitar los procesos cognitivos humanos que permiten interactuar con los distintos entornos [Koo et al., 2011].

Tal como se ha visto, los robots afectivos mejoran el proceso de aprendizaje de los alumnos. No obstante, para operar como un auténtico tutor y realizar las intervenciones pedagógicas oportunas, un robot debe conocer la materia de aprendizaje, el nivel de competencia de los alumnos, y las circunstancias concretas de aprendizaje que demanda cada particular intervención. Una posible medio de proporcionar a un robot este conocimiento consiste en establecer una comunicación entre el robot y un sistema recomendador educativo afectivo con el que los alumnos interactúan.

En el contexto de la asistencia médica se han realizado ya avances en cuanto a comunicación entre robots y sistemas recomendadores. Es así como en [Hammer et al., 2015] se ha desarrollado un sistema llamado CARE que provee recomendaciones para ayudar a sus usuarios a superar las dificultades del día a día y mejorar su bienestar. En este estudio se implementó CARE en diferentes dispositivos, entre ellos un robot.

Al igual que en [Hammer et al., 2015], en [Tang et al., 2015] se presenta un robot que asiste a personas adultas. Este robot tiene la capacidad de conectarse con un sistema recomendador con el propósito de mejorar la asistencia a estas personas y también sus conversaciones con ellas.

Al igual que en el contexto de la asistencia médica, también se han desarrollado robots conectados a sistemas recomendadores en el contexto de la prestación de otros servicios como se muestra en [Koo et al., 2011], donde un robot realiza una serie de recomendaciones sobre cafés en base a las preferencias de los usuarios.

Dentro asimismo del contexto de servicios, en [Kamei et al., 2011] se propone la conexión de un robot a un sistema recomendador dentro de una tienda, para que, basado en los intereses del usuario realice a los clientes las recomendaciones oportunas. Con la introducción de robots recomendadores se consigue que los usuarios estén más satisfechos con los servicios, encontrando antes los productos deseados, e incluso descubriendo nuevos productos adecuados a sus preferencias.

Sin embargo, en el contexto de la educación no se han encontrado hasta la fecha estudios donde se relacione un sistema recomendador educativo con un robot.

CAPÍTULO 3

ARQUITECTURA ASIART

En este capítulo se describe la arquitectura propuesta para la implementación de robots tutores afectivos que operen asociados a un software educativo.

3. Arquitectura ASIART

En este capítulo se describen los diferentes componentes de la arquitectura ASIART.

La arquitectura ASIART está enmarcada dentro de una arquitectura de robot reactiva-deliberativa, que implica los paradigmas situado, conexionista y simbólico. El robot interactúa con su entorno en tiempo real, identificando a los alumnos mediante sensores, y dialogando y actuando con ellos mediante efectores, en función de sus percepciones cambiantes tanto en forma reactiva como deliberativa.

3.1. Arquitectura general

En la figura 3.1 se muestra el patrón arquitectónico de ASIART.

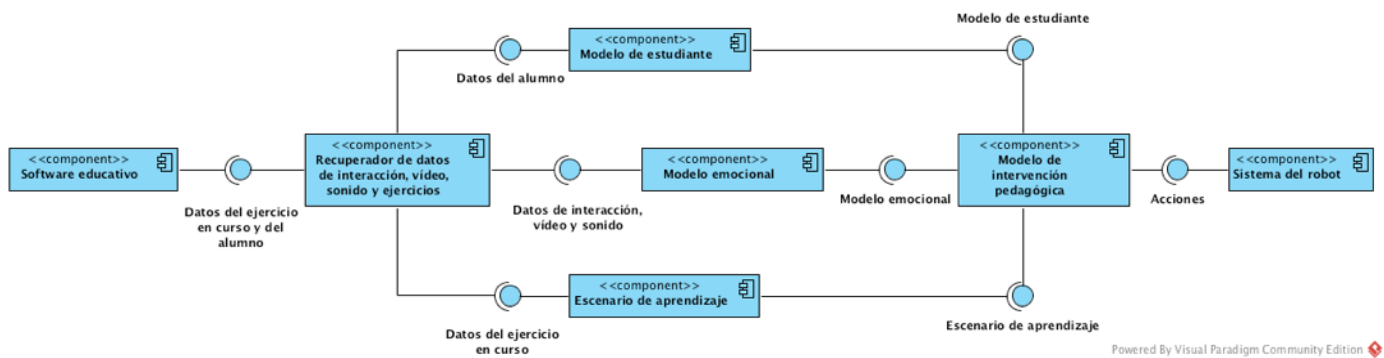


Figura 3.1: Patrón arquitectural de ASIART

Esta arquitectura está enmarcada dentro de una arquitectura de robot reactiva-deliberativa.

- **Software educativo:** Este es el software con el que el alumno interactúa.
- **Recuperador de datos de interacción, vídeo, sonido y actividades de aprendizaje:** Este componente recibe del software educativo los datos de la actividad de aprendizaje en curso y los datos de interacción, de vídeo y de sonido del alumno que está realizando la actividad. Dentro de la arquitectura reactiva-deliberativa del robot, este componente forma parte del sistema de percepción, ya que es donde se obtiene la información dinámica del alumno (datos de interacción, vídeo y sonido) y de las actividades que está realizando. Este componente contiene también una parte de la memoria a corto plazo, donde se registran los datos variables de las actividades, las interacciones, el vídeo y el sonido.
- **Modelo de estudiante:** Este componente recibe los datos de evaluación de la actividad del alumno, y devuelve un modelo de estudiante basado en el historial académico y características personales (estilo de aprendizaje, competencias, etc). Dentro de la arquitectura reactiva-deliberativa del robot, este componente forma parte del razonador, ya que con los datos de evaluación de la actividad del alumno recibidos actualiza un modelo dinámico que sirve para establecer un plan de acción. Posee una parte de la memoria a largo plazo dentro de la arquitectura del

robot, en donde se almacena la información sobre el historial académico del alumno y sus características personales.

- **Modelo emocional:** Este modelo recibe los datos de interacción, vídeo y sonido, y devuelve un modelo emocional basado en las interacciones de teclado y ratón, las características extraídas del vídeo (análisis facial, gestual, etc) y las características extraídas del sonido (análisis de voz, análisis de ruido ambiente, etc). Dentro de la arquitectura reactiva-deliberativa del robot, este componente forma parte del sistema perceptivo y parte del razonador puesto que, en base a percepciones, identifica el estado emocional que determinará el plan de acción (en forma reactiva o deliberativa, según los casos). Este componente contiene también una parte de la memoria a corto plazo, donde se almacenan las interacciones, el vídeo y el sonido recuperados.
- **Escenario de aprendizaje:** Este componente recibe la información relativa a la actividad de aprendizaje que se está realizando y devuelve una especificación del escenario de aprendizaje.
- **Modelo de intervención pedagógica:** Este modelo recibe el modelo dinámico de estudiante, el estado emocional y el escenario de aprendizaje, y devuelve las acciones que tiene que realizar el robot.
- **Sistema del robot:** Dentro de la arquitectura reactiva-deliberativa, esta parte corresponde al sistema de ejecución que se encarga de ordenar los movimientos en los actuadores. Este componente contiene una parte de la memoria a corto plazo, donde se almacena la identificación del alumno en cuestión, así como la información obtenida mediante los sensores del robot.

CAPÍTULO 4

EXPERIENCIA DE USUARIO CON SCRATCH

En este capítulo se detallan los procedimientos seguidos para obtener los datos que han servido para el desarrollo del modelo emocional y el modelo de intervención pedagógica.

4. Experiencia de usuario con Scratch

La experiencia que a continuación se describe tuvo como objetivos la recogida de datos para el desarrollo del modelo emocional, y la elicitación de conocimiento sobre intervención pedagógica afectiva para el desarrollo del modelo de intervención pedagógica.

Tal como se ha indicado en capítulos anteriores, el reconocimiento de estados cognitivo-afectivos se ha basado en el análisis de las interacciones de teclado y ratón, método no intrusivo y de bajo coste que proporciona buenos resultados en el caso que nos ocupa. Puesto que este proyecto está orientado a alumnos de primaria y no se encontraron referencias, para este tipo de alumnos, de bancos de datos de interacciones de teclado y ratón etiquetados con caracterizaciones de estado cognitivo-afectivo, se presentó la necesidad de crear uno mediante sesiones de grabación.

Las sesiones de grabación se realizaron en el colegio Los Peñascales de Las Rozas (Madrid). Se contó con dos grupos de 10 alumnos cada uno, compuestos por niños y niñas con edades comprendidas entre los 10 y 11 años (5º de primaria). Cada uno de los grupos participó en dos sesiones de grabación de 45 minutos por sesión. En las sesiones de grabación se situó una cámara de vídeo fija en una de las esquinas del aula, de forma que se grabara toda la escena y poder recoger las respuestas pedagógicas afectivas de la profesora que colaboró en la experiencia. Para recoger los datos de las interacciones de teclado y ratón de cada uno de los ordenadores con los que trabajaban los alumnos se utilizó un software específico.

A la hora de seleccionar este software específico para la recogida de datos de las interacciones de teclado y ratón, se tomaron en consideración los siguientes requisitos:

- El registro de las interacciones de teclado y ratón había de estar en un formato estándar. Para ello se decidió almacenar dichas interacciones en el formato CSV con separador de “;”.
- A la hora de registrar el log, debían tomarse en cuenta las interacciones tanto con Scratch, como con cualquier otro software (navegadores, juegos, etc) para detectar posibles comportamientos que no correspondieran a las interacciones normales con Scratch. Esto hizo que este registro tuviera que ser independiente del software educativo.
- Puesto que los logs tenían que ser analizados y etiquetados a posteriori, era necesario contar con soportes auxiliares que ayudaran a la identificación de cada uno de los estados cognitivo-afectivos. Esto llevó a realizar registros de las grabaciones obtenidas a través de las webcams de los ordenadores, así como de videos de captura de pantalla para identificar las acciones realizadas por los alumnos en cada momento. De esta forma se podrían identificar los estados cognitivo-afectivos de los alumnos y analizar adicionalmente las interacciones con el ordenador.

Como se mencionó anteriormente, se realizaron también grabaciones generales de las sesiones que permitieran estudiar las intervenciones de la profesora en función de la conducta y las dudas de los alumnos y tomarlas como ejemplo para la programación del modelo de intervención pedagógica.

Además, como apoyo a las grabaciones, se contó con una serie de observadores externos que tomaron notas en un formulario en papel, diseñado según la metodología BROMPT.

Debido a que no fue posible encontrar un software que se adaptase a los requisitos especificados sobre el registro de logs y videos, se decidió realizar un software a medida para estos propósitos. El software se diseñó para que fuera independiente de Scratch y parametrizable, y así poder ser reutilizado con cualquier otra herramienta software educativa. No obstante, al tratarse de un software de registro de interacciones de teclado y ratón, la herramienta educativa que haga uso de este software deberá ser una herramienta que se opere principalmente a través de teclado y de ratón.

El software fue desarrollado en Java, de manera que se pudiera ejecutar en cualquier sistema operativo, y cuenta con una interfaz sencilla para facilitar el uso a los niños de primaria. Esta interfaz se puede ver en la figura 4.1.

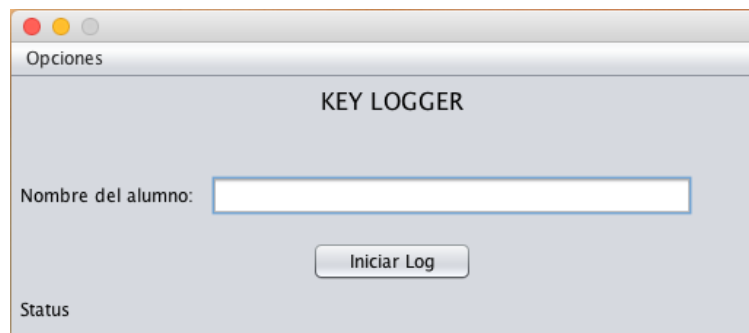


Figura 4.1: Interfaz del software de registro de logs y vídeos

En esta interfaz se indica el nombre del alumno (o su identificación), y al pulsar sobre el botón "iniciar Log" se genera un fichero csv donde se registrarán el log y los correspondientes ficheros de video (uno para la webcam y otro para la captura de pantalla). Para identificar fácilmente a quién corresponde cada uno de los ficheros, estos ficheros se nombran con el identificador del alumno que se ha indicado en la interfaz. El único botón de la interfaz sirve tanto para iniciar la grabación, como para detenerla (viene etiquetado como "Iniciar Log" y "Detener Log").

Esta aplicación ha permitido la recogida de datos y vídeos de las 4 sesiones de grabación que posteriormente han sido analizados.

En las figuras 4.2 y 4.3 se muestran dos fotogramas de las sesiones realizadas.



Figura 4.2: Fotograma de la experiencia de usuario con Scratch



Figura 4.3: Fotograma de la experiencia de usuario con Scratch

CAPÍTULO 5

METODOLOGÍA ASIART

En este capítulo se detalla la metodología para la adaptación e implementación de ASIART.

5. Metodología ASIART

A continuación se presenta brevemente un conjunto de directrices y fases aplicables para la implementación de robots tutores afectivos en el contexto de la arquitectura ASIART:

- *Primera fase: Experiencia de usuario.* La elicitación del conocimiento sobre intervenciones pedagógicas es un proceso básico de cara diseñar la estrategia de tutorización afectiva que seguirá el robot cuando interactúe con los alumnos. Para esta elicitación se realizan una o varias experiencias de usuario con el software educativo en cuestión, con el fin de obtener información acerca de cómo y cuándo actúa el profesor, y bajo qué circunstancias, es decir, qué escenarios de uso del software educativo requieren con más probabilidad de intervención pedagógica.
 - Para empezar, es necesario diseñar una serie de actividades para llevar a cabo con el software educativo tales que susciten en los alumnos el mayor número posible de estados emocionales que se deseen analizar.
 - En lo que respecta al registro de las experiencias, las sesiones se tienen que grabar en un formato tal que facilite el posterior análisis de las diferentes situaciones que los alumnos experimenten, y de las actuaciones del profesor (generalmente en formato de audio y vídeo). Como complemento a estas grabaciones, durante las sesiones puede también contarse con observadores que cumplimenten formularios basados en la metodología BROMPT, realizando anotaciones sobre las intervenciones pedagógicas, y las actitudes y respuestas emocionales de los alumnos que adviertan durante la experiencia.
 - Por último, en el caso de que se desee generar un modelo de clasificación de estados emocionales específico, será necesario desarrollar herramientas para la recopilación de los datos que serán analizados para la posterior generación del modelo.
- *Segunda fase (opcional): Generación del modelo de clasificación de estados emocionales mediante técnicas de aprendizaje computacional o minería de datos* (recordamos que en este proyecto se ha desarrollado un modelo de clasificación de estados emocionales utilizable con diferentes programas educativos).
- *Tercera fase: Instanciación del modelo de intervención pedagógica.* Después de analizar las grabaciones y anotaciones de las experiencias se parametriza el modelo de intervención pedagógica (explicado en el siguiente capítulo) a través de un framework que sirve para facilitar dicha tarea al usuario.
- *Cuarta fase: Desarrollo de un módulo de identificación de escenarios educativos.* Dependiendo del software educativo que se vaya a utilizar y de los contenidos de la materia objeto de aprendizaje, será necesario implementar un módulo de identificación de escenarios educativos.
- *Quinta fase (opcional): Desarrollo de un traductor de intervenciones pedagógicas a acciones del robot.* Finalmente, dependiendo del robot que se quiera utilizar, se tendría que implementar un traductor de las acciones determinadas por el modelo de intervenciones pedagógicas al robot en cuestión. En el caso de que el robot utilizado fuera NAO, la plataforma ASIART proporciona ya tal traductor.

CAPÍTULO 6

DESARROLLO DEL MODELO EMOCIONAL

En este capítulo se detallan los procesos de análisis de datos, desde la selección de las características más relevantes, hasta la creación del modelo emocional.

6. Desarrollo del modelo emocional

6.1. Módulo de análisis de logs

Una vez recopilados y posteriormente analizados los ficheros log y de vídeo de todas las sesiones se concluyó que existían tres estados cognitivo-afectivos diferenciables de forma sencilla y que condicionaban significativamente la naturaleza de la intervención pedagógica de la profesora : “Concentrado”, “Desconcentrado” e “Inactivo”. En el estado “Concentrado”, el alumno está realizando las tareas encomendadas en Scratch, en el estado “Desconcentrado”, el alumno está realizando otras tareas en el ordenador que no han sido encomendadas o bien realizando acciones aleatorias, y en el estado “Inactivo” el alumno no está interactuando con el ordenador, bien porque tiene una duda y está preguntando al profesor, bien porque está distraído hablando con un compañero. La interpretación de estos estados en clave cognitivo-afectiva se describió más extensamente en la sección 2.2.4.

Después de determinar los diferentes estados cognitivo-afectivos de cada uno de los alumnos participantes estudiando las grabaciones de video y las notas tomadas por los observadores, se etiquetó cada una de las líneas de los log con el estado correspondiente. Al resultar los estados bastante básicos y sencillos de determinar mediante la visualización de los vídeos, no fue necesaria la intervención de un experto.

Dado que el proceso de etiquetado de cada una de las líneas de los ficheros log habría supuesto un proceso demasiado costoso en tiempo, se decidió buscar un software que simplificase esta tarea. Para ello, dicho software tenía que cumplir los siguientes requisitos:

- Reproducir, pausar y avanzar de forma síncrona el vídeo de la webcam y el vídeo de la captura de pantalla.
- Etiquetar las líneas del log por franjas de tiempo (minutos y segundos).
- Exportar el resultado a un fichero CSV con separador de “;”.

Debido a los requisitos tan específicos de este software, no fue posible encontrar ninguna herramienta que los reuniera, por lo que se decidió desarrollarla. Al igual que la herramienta descrita en el apartado anterior para la recogida de datos, fue desarrollada en JAVA. La interfaz de esta herramienta de etiquetado de estados puede verse en las figuras 6.1 y 6.2:



Figura 6.1: Interfaz principal del software de etiquetado

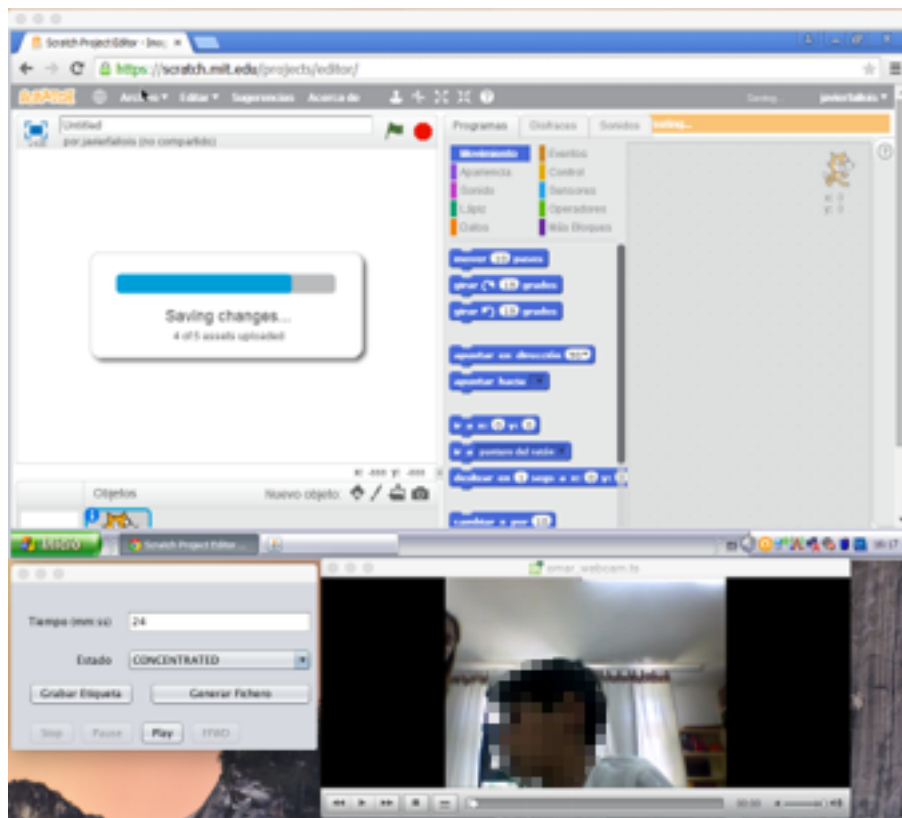


Figura 6.2: Interfaz de reproducción del software de etiquetado

Tal como se ha descrito en la sección 2.2 del capítulo del estado del arte ("Reconocimiento de estados emocionales a partir de análisis de logs"), los parámetros de interacción de teclado y de ratón analizados para la generación del modelo de clasificación fueron:

- Número de pulsaciones sobre la tecla de borrado (backspace)
- Número de pulsaciones sobre la tecla de suprimir
- Latencia de teclado
- Tiempo de presión de las teclas
- Número de clicks de ratón
- Distancia media de los clicks de ratón
- Número de pausas de ratón
- Tamaño de pausas de ratón
- Número de movimientos bruscos de ratón
- Velocidad máxima del ratón
- Velocidad mínima del ratón

- Velocidad media del ratón

Ya que para esta labor necesaria, al igual que ocurrió en las fases anteriores, se requiere un software que satisfaga unos requisitos muy concretos, se ha desarrollado una herramienta que realiza el cálculo de los parámetros anteriores en base a las entradas del log y las etiquetas de estados cognitivo-afectivos.

Los parámetros tienen que estar referidos a una unidad de tiempo concreta por lo que, contando con que las sesiones eran de 45 minutos y se dieron cambios rápidos entre los diferentes estados cognitivo-afectivos, se decidió realizar las mediciones por minuto. Esta herramienta, al igual que las anteriores, se desarrolló en JAVA. Su interfaz puede verse en la figura 6.3.



Figura 6.3: Interfaz del software de análisis de datos

Finalmente, puesto que los datos se iban a analizar con la herramienta Weka, se decidió establecer el formato del fichero de salida en ARFF (Attribute-Relation File Format), de manera que fuera completamente compatible con Weka.

6.1.Descripción de la muestra

En este proyecto se ha trabajado con una muestra de las interacciones de teclado y ratón de 20 alumnos, de entre 10 y 11 años, en 4 sesiones de 45 minutos cada sesión. El tamaño del fichero de log de cada alumno varía dependiendo del tiempo real de la sesión y del número de interacciones que haya realizado durante la misma. A continuación se muestra una tabla resumen con el número de instancias por estado de concentración del alumno.

Estado de concentración	Número de instancias
Concentrado	670
Distraído	89
Inactivo	600
TOTAL	1359

Tabla 6.1: Datos de muestra por estado de concentración

6.2. Selección de atributos

La selección de atributos tiene cuatro objetivos principales. En primer lugar, permite reducir el tamaño de los datos. En segundo lugar, puede mejorar la calidad del modelo, al hacer que éste se centre en las características relevantes. En tercer lugar, al tener menos atributos mejora la lectura de los modelos, tales como árboles de decisión, regresión lineal, etc. Por último también facilita la comprensión de los datos de forma visual [Hernández Orallo et al., 2004]. En nuestro caso, el objetivo más relevante es el de poder mejorar la calidad del modelo eliminando las características menos relevantes.

La tabla 6.2 representa los atributos que se han utilizado como entrada para los métodos de selección y evaluación.

ID	Atributos	Descripción
1	BackspaceKeys	Número de pulsaciones de la tecla de borrado
2	DeleteKeys	Número de pulsaciones de la tecla "Suprimir"
3	LatencyKeys	Latencia de tecleo
4	PresstimeKeys	Tiempo de presión de las teclas
5	ClicksMouse	Número de clicks de ratón
6	MeanDistanceClicksMouse	Distancia media de los clicks de ratón en píxeles
7	PauseMovementMouse	Número de las pausas en el movimiento de ratón
8	PauseMovementSizeMouse	Tamaño de las pausas en el movimiento de ratón
9	AbruptMovementMouse	Número de eventos "bruscos" de ratón
10	MaxSpeedMouse	Velocidad máxima de movimiento de ratón
11	MinSpeedMouse	Velocidad mínima de movimiento de ratón
12	MeanSpeedMouse	Velocidad media de movimiento de ratón
13	CLASS	Clasificación de la muestra

Tabla 6.2: Atributos de entrada para los métodos de selección y evaluación

Dentro de los métodos de selección de atributos existen dos tipos diferentes de métodos: los métodos de selección de atributos, que nos proporcionan un conjunto básico de atributos significativos, y los métodos de evaluación que nos proporcionan un ranking de atributos de más a menos significativos. Ambos tipos de métodos son complementarios. Puesto que los atributos que se manejan son continuos, se han seleccionado dentro de las posibilidades, aquellos métodos que resultan tener un buen comportamiento con este tipo de datos. Estos métodos se describen a continuación.

6.2.1. Métodos de selección

Además de los métodos de selección más comunes (CFS, Consistency, CostSensitive, etc), dentro de los métodos de selección también existen unos métodos basados en modelos. Estos métodos devuelven un subconjunto de atributos óptimos que evalúa a

basa de ejecutar métodos de aprendizaje. En nuestro caso se han escogido tres de éstos métodos que funcionan bastante bien con atributos continuos: Árbol C4.5, Redes Bayesianas y las Máquinas de Vectores Soporte.

A. Selección de atributos por Correlación o CFS (Correlation Feature Selection)

$$CFS = \max_{sk} \left[\frac{r_{cf1} + r_{cf2} + \dots + r_{cfk}}{\sqrt{k+2(r_{f1f2} + \dots + r_{f1fk} + \dots + r_{fkf1})}} \right] \quad (6.1)$$

En la tabla 6.3 *CfsSubsetEval*. Este método genera un conjunto de atributos óptimos basándose en la capacidad de cada uno de estos atributos de determinar la clase, así como de su redundancia. El método de búsqueda utilizado es *BestFirst*, que realiza una búsqueda del espacio de subconjuntos de atributos por hillclimbing codicioso.

Siendo r_{cfi} y r_{fijj} las variables que hacen referencia a las correlaciones.

Esto nos proporciona por tanto qué atributos tienen mayor capacidad de clasificar por sí mismos, calculando la correlación existente entre cada atributo y la clase, y eliminando tanto los atributos que tengan una correlación muy alta, como atributos redundantes (en los que exista una correlación entre atributos muy alta).

B. Árbol C4.5

En la tabla 6.3 *J48*. En cada nodo del árbol se escoge el atributo que divida el conjunto de datos de la forma más eficaz. Este método se basa en la ganancia de información (diferencia de entropía) para realizar la selección del atributo como parámetro de decisión. La ejecución de este método se realizará sin poda, de forma que no se controlará el sobreajuste.

Algoritmo 6.1 C4.5

Entrada: dataset T ; atributos S .

Salida: árbol de decisión *Tree*.

```

if  $T$  es NULL then
    return failure
end if
if  $S$  es NULL then
    return Tree como nodo único con la etiqueta de la clase más frecuente en  $T$ .
end if
if  $|S| = 1$  then
    return Tree como un sólo nodo  $S$ .
end if
set  $Tree = \{\}$ 
for  $a \in S$  do
    establece  $Info(a, T) = 0$ , and  $SplitInfo(a, T) = 0$ 
    calcula  $Entropy(a)$ 

```

for $v \in \text{values}(a, T)$ **do**

establece $T_{a,v}$ como el subset de T con atributo $a = v$

$$\text{Info}(a, T) += \frac{|T_{a,v}|}{|T_a|} \text{Entropía}(a_v)$$

$$\text{SplitInfo}(a, T) += - \frac{|T_{a,v}|}{|T_a|} \log \frac{|T_{a,v}|}{|T_a|}$$

end for

$$\text{Gain}(a, T) = \text{Entropy}(a) - \text{Info}(a, T)$$

$$\text{GainRatio}(a, T) = \frac{\text{Gain}(a, T)}{\text{SplitInfo}(a, T)}$$

end for

establece $a_{\text{best}} = \text{argmax}\{\text{GainRatio}(a, T)\}$

inserta a_{best} en *Tree*

for $\text{values}(a_{\text{best}}, T)$ **do**

llama a C4.5($T_{a,v}$)

end for

return *Tree*

Siendo C el número de clases, y $p(S, j)$ la proporción de instancias en S que son asignadas a la clase número j . Por lo tanto, la entropía del atributo S se calcula como:

$$\text{Entropy}(S) = - \sum_{j=1}^C p(S, j) \times \log p(S, j) \quad (6.2)$$

En consecuencia, la ganancia de la información por un dataset de entrenamiento T se define como:

$$\text{Gain}(S, T) = \text{Entropy}(S) - \sum_{v \in \text{values}(T_s)} \frac{|T_{S,v}|}{|T_S|} \text{Entropy}(S_v) \quad (6.3)$$

donde $\text{values}(T_s)$ es el set de valores de S en T , T_s es el subset de T inducido por S , y $T_{S,v}$ es el subset de T en cuyo atributo S tiene un valor de v .

Por lo tanto, el ratio de ganancia de información del atributo S se define como:

$$GainRatio(S,T) = \frac{Gain(S,T)}{SplitInfo(S,T)} \quad (6.4)$$

donde SplitInfo(S,T) se calcula como:

$$SplitInfo(S,T) = - \sum_{v \in values(T_s)} \frac{|T_{s,v}|}{|T_s|} \times \log \frac{|T_{s,v}|}{|T_s|} \quad (6.5)$$

C. Redes bayesianas

En la tabla 6.3 *BayesNet*. Este método está basado en el teorema de Bayes.

$$P(h|O) = \frac{P(O|h)P(h)}{P(O)} \quad (6.6)$$

donde P(h) son las probabilidades a priori, P(O|h) es la probabilidad de observación (probabilidad de O en la hipótesis h) y P(h|O) son las probabilidades a posteriori.

En las redes bayesianas se crea una red, donde cada uno de los nodos tiene una probabilidad condicional a los padres de que se cumpla. Para generar esta red se ha usado un algoritmo K2, el cual utiliza un esquema voraz en su búsqueda de soluciones candidatas cada vez mejores, y parte de que las variables de entrada estén ordenadas [Hernández Orallo et al., 2004].

Algoritmo 6.2 K2

Entrada: nodos X (variables ordenadas); datos D

```

for  $i \in n$  do
    establece  $X_i = 0.0$ 
end for
for  $i \in n$  do
    establece  $ok = true$ 
    do while  $ok$ 
        sea  $X_j$  el nodo tal que  $j < i$  y  $X_j$  no pertenezca a  $Pa(X_i)$  que maximiza
         $f_i(X_i | Pa(X_i) \cup X_j : D)$ 

        if  $f_i(X_i | Pa(X_i) \cup X_j : D) > f_i(X_i | Pa(X_i) : D)$  then
            establece  $Pa(X_i) = Pa(X_i) \cup X_j$ 

        else
            establece  $ok = false;$ 
        end if
    end while
end for

```

D. Máquinas de Vectores Soporte o SVM (Support Vector Machines)

En la tabla 6.3 *SVM*. Las máquinas de vectores soporte o SVM (Support Vector Machines) pertenecen a la familia de los clasificadores lineales, puesto que inducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad. La idea de las SVM consiste en seleccionar el hiperplano separador que está a la misma distancia de los ejemplos más cercanos de cada clase [Hernández Orallo et al., 2004].

Por lo que suponiendo que el conjunto de datos es linealmente separable en el espacio de entrada tendremos que, siendo α^* una solución del problema dual:

Maximizar

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \quad (6.7)$$

Sujeto a

$$\sum_{i=1}^N y_i \alpha_i = 0 \quad (6.8)$$

$$\alpha_i \geq 0 \quad 1 \leq i \leq N$$

Entonces, el vector

$$\omega^* = \sum_{i=1}^N y_i \alpha_i^* x_i \quad (6.9)$$

es el vector ortogonal al hiperplano con margen (geométrico) máximo. La SVM lineal con margen máximo es:

$$h(x) = \langle \omega^*, x \rangle + b^* = \sum_{i=1}^N y_i \alpha_i^* \langle x_i, x \rangle + b^* \quad (6.10)$$

donde

$$b^* = -\frac{1}{2} (\max_{y_j=-1} \{ \langle \omega^*, x_j \rangle \} + \max_{y_j=+1} \{ \langle \omega^*, x_j \rangle \}) \quad (6.11)$$

y su clasificador asociado es

$$f(x) = \text{signo}(h(x)) \quad (6.12)$$

Este método además dispone de varios tipos de kernel diferentes siendo los más comunes el kernel polinómico:

$$K(x,y)=\langle x,y \rangle^p \quad (6.13)$$

Y el kernel RBF (Radial Basis Function):

$$K(x,y)=e^{-(\text{gama}*\langle x-y,x-y \rangle^2)} \quad (6.14)$$

6.2.2. Métodos de evaluación

Para los métodos de evaluación se ha utilizado *Rank* como método de búsqueda, el cual evalúa los atributos de manera individual.

A. Ganancia de Información o IG (Information Gain)

En la tabla 6.3 *InformationGainAttributeEval*. La ganancia de información o IG (Information Gain) evalúa el valor de un atributo mediante la medición de ganancia de información con respecto a la clase [Irani and Fayyad, 1993].

La entropía es una medida considerada como una medida de la incertidumbre del sistema. La entropía de Y es:

$$H(Y)=-\sum_{y \in Y} p(y) \log_2(p(y)) \quad (6.15)$$

donde $p(y)$ es la función de la densidad de la probabilidad marginal para la variable aleatoria Y.

Si los valores observados de Y en los datos de entrenamiento S se dividen de acuerdo a los valores de una segunda característica X, y la entropía de Y con respecto a las particiones indicadas por X es menor que la entropía de Y antes de la partición, entonces hay una relación entre características X e Y.

La entropía de Y después de observar X es la siguiente:

$$H(Y|X)=-\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2(p(y|x)) \quad (6.16)$$

donde $p(y|x)$ es la probabilidad condicional de y dado x.

Teniendo en cuenta la entropía como criterio de impureza en un dataset de entrenamiento S, podemos definir una medida que refleja más información sobre Y provista por X que representa la cantidad por la cual la entropía de Y disminuye.

Esta medida se conoce como IG que es dada por:

$$IG=H(Y)-H(Y|X)=H(X)-H(X|Y) \quad (6.17)$$

IG es por tanto una medida simétrica en la que la información obtenida sobre Y después de observar X es igual a la información obtenida sobre X después de observar Y.

B. ReliefF

En la tabla 6.3 *ReliefFAttributeEval*. Evalúa el valor de un atributo por muestreo repetitivo de una instancia y teniendo en cuenta el valor del atributo dado por el más cercano de la misma clase y el más cercano de diferente clase [Kononenko, 1994].

El algoritmo que define este método es el siguiente:

Algoritmo 6.3 ReliefF

establece todos los pesos $W_A=0.0$

for $i \in m$ **do**

 selecciona de forma aleatoria una instancia R

 busca el acierto más cercano H y el fallo más cercano M

for A \in todos los atributos **do**

$W_A = W_A - \text{diff}(A,R,H)/m + \text{diff}(A,R,M) / m$

end for

end for

En el caso de ReliefF:

$$W_A = W_A - \frac{\text{diff}(A,R,H)}{m} + \sum_{c \neq \text{class}(R)} \frac{[P(C)\text{diff}(A,R,M(C))]}{m} \quad (6.18)$$

6.2.3. Resultados de la selección y evaluación de atributos

A continuación, se muestran la tabla resultante de la ejecución de los diferentes métodos de selección y evaluación de atributos:

Método Selección	Filtro/Modelo	Opciones	Estrategia	Atributos seleccionados
CfsSubsetEval	Filtro	Valores por defecto	BestFirst	1,2,4,5,6,7,8,9
WrapperSubsetEval	Modelo	J48	BestFirst	2,3,4,5,6
		BayesNet	BestFirst	3,4,5,6,9,12
		SMO	BestFirst	1,4,5,6,7,9,10,11
Método Evaluación	Filtro/Modelo		Estrategia	Orden de los atributos
InfoGainAttributeEval	Filtro		Ranker	7,5,8,10,12,4,3,9,6,1,11,2
ReliefFAttributeEval	Filtro		Ranker	8,7,3,4,6,12,10,1,5,9,2,11

Tabla 6.3: Tabla de resultados de la selección y evaluación de atributos

El atributo *MinSpeedMouse*, aunque aparece en el método de selección *SMO* no es un atributo relevante, ya que en cualquier estado de concentración la velocidad mínima del ratón puede ser cero (por ejemplo, si el alumno está escribiendo y el ratón no se está moviendo). Se puede comprobar en los métodos de evaluación que este parámetro (11) aparece en última o penúltima posición de relevancia.

Dos atributos que no resultan relevantes en este estudio son los que indican las pulsaciones de teclas concretas (*BackspaceKeys* (1), *DeleteKeys* (2)), ya que la pulsación de estas teclas no dice nada sobre el estado de concentración, y las pulsaciones se pueden leer tanto con el atributo *LatencyKeys* (3), como con el atributo *PressTimeKeys* (4). En el método de evaluación *InfoGainAttributeEval* se puede observar que estos atributos (1 y 2) están al final del ranking.

ID	Atributos	Descripción
3	LatencyKeys	Latencia de tecleo
4	PresstimeKeys	Tiempo de presión de las teclas
5	ClicksMouse	Número de clicks de ratón
6	MeanDistanceClicksMouse	Distancia media de los clicks de ratón en píxeles
7	PauseMovementMouse	Número de las pausas en el movimiento de ratón
8	PauseMovementSizeMouse	Tamaño de las pausas en el movimiento de ratón
9	AbruptMovementMouse	Número de eventos “bruscos” de ratón
10	MaxSpeedMouse	Velocidad máxima de movimiento de ratón
12	MeanSpeedMouse	Velocidad media de movimiento de ratón
13	CLASS	Clasificación de la muestra

Tabla 6.4: Selección de atributos de la muestra de datos

En la tabla 6.4 se muestran los atributos que finalmente se utilizarán para los métodos de clasificación y agrupamiento.

6.3. Modelo de clasificación

Tal como se ha comentado en el capítulo 2.2.5 del estado del arte, por su buen comportamiento en experimentos similares se van a analizar los siguientes métodos de clasificación: K-vecinos, C4.5, Redes Neuronales Artificiales y Naïve Bayes.

Además de estos métodos, también se va a analizar SVM, ya que tiene un buen comportamiento con atributos continuos, y trabaja bien con alta dimensionalidad realizando separaciones con hiperplanos.

A excepción de los métodos C4.5 y SVM definidos ya en el capítulo 6.2.1 de la selección de atributos, el resto de métodos se van a definir a continuación.

6.3.1. Naïve Bayes

El fundamento principal de Naïve Bayes (o NB) es la suposición de que todos los atributos son independientes conocido el valor de la variable clase. Esta hipótesis de independencia asumida por el clasificador NB da lugar a un modelo gráfico probabilístico en el que existe un único nodo raíz (la clase), y en el que todos los atributos son nodos hoja que tienen como único nodo padre a la variable clase [Hernández Orallo et al., 2004].

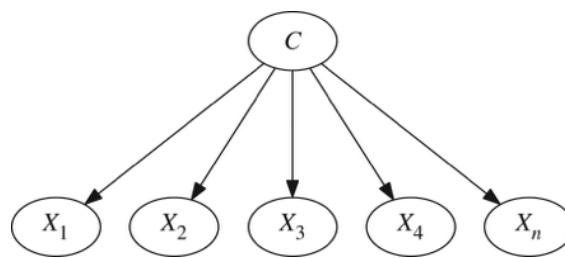


Figura 6.4: Topología de un Clasificador Naïve Bayes

Debido a la hipótesis de independencia usada en Naïve Bayes, la expresión para obtener la hipótesis MAP queda como sigue:

$$c_{MAP} = \arg \max_{c \in \Omega_c} p(c) \prod_{i=1} p(A_i | c) \quad (6.19)$$

Por tanto, los parámetros que hay que estimar son $P(A_i | c)$ para cada atributo y la probabilidad a priori de la variable clase $P(c)$.

Dependiendo de si el atributo A_i es discreto o continuo, la estimación se realiza de forma diferente. En nuestro caso, los atributos son todos continuos, por lo que el clasificador NB supone que los atributos siguen una distribución normal, y lo único que habría que calcular es la media μ y la desviación típica σ condicionadas a cada valor de la variable clase.

$$P(A_i | c) \propto N(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(X-\mu)^2}{2\sigma^2}\right) \quad (6.20)$$

6.3.2. Redes Neuronales Artificiales

En este estudio se utilizará el Perceptrón Multicapa (o MLP por sus siglas en inglés). El MLP resulta un método potente de aprendizaje, en el que conjuntos de datos que no son linealmente separables pueden ser modelados mediante este método [Hernández Orallo et al., 2004]. Este método es una red neuronal en forma de cascada, que tiene una o más capas ocultas, tal como se muestra en la figura 6.5.

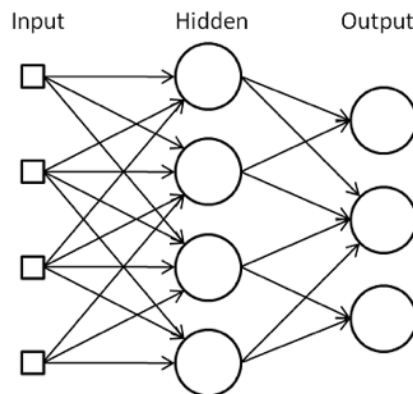


Figura 6.5: Topología de un perceptrón multicapa

En esta red, la activación se propaga a través de los pesos que hay entre la capa de entrada y la capa oculta, y entre la capa oculta y la capa de salida. Por lo tanto, en el aprendizaje hay que actualizar estos dos conjuntos de pesos.

El error debido al primer conjunto de pesos se calcula empleando el método del error cuadrático medio. Siendo γ el coeficiente de aprendizaje, tenemos que el error cuadrático medio se define como:

$$\Delta_P w_j = -\gamma \frac{\partial E^P}{\partial w_j} \quad (6.21)$$

Una vez calculado el error cuadrático medio, se propaga hacia atrás la parte del error debida a los errores que tienen lugar en el segundo conjunto de pesos y se asigna el error proporcional a los pesos que lo causan.

Algoritmo 6.4 Retropropagación

1. Inicializar los pesos a valores pequeños aleatorios.
2. Escoger un patrón de entrada, x^P , y presentarlo a la capa de entrada.
3. Propagar la activación hacia delante a través de los pesos hasta que la activación alcance las neuronas de la capa de salida.
4. Calcular los valores de "o" para las capas de salida usando:

$$o_j^P = (t_j^P - o_j^P) f'(Act_j^P)$$

5. Calcular los valores de “ δ ” para la capa oculta usando:

$$\delta_j^P = \sum_{j=1}^N \delta_j^P w_{ji} f'(Act_j^P)$$

6. Actualizar los pesos de acuerdo con:

$$\Delta_P w_{ij} = \gamma \delta_i^P o_j^P$$

7. Repetir del paso 2 al 6 para todos los patrones de entrada.

En este algoritmo, tenemos que $f'(Act_j^P)$ representa la derivada de la función de activación.

En nuestro caso se utilizará una función de activación sigmoide, que se define como:

$$f(s) = \frac{1}{1 + \exp(-bx)} \quad (6.22)$$

Por lo que tendremos:

$$f'(s) = b \cdot (1 - f(x)) \cdot f(x) \quad (6.23)$$

El principal problema del método de retropropagación básico es su lentitud de convergencia, sin embargo, si se incrementa la velocidad de convergencia demasiado, se introduce inestabilidad en la regla de aprendizaje. Lo más simple para aumentar la velocidad es aplicar un termino llamado momento, al cambio de los pesos. La idea básica es incrementar si el nuevo cambio de pesos coincide con la dirección de los cambios anteriores y reducir en caso contrario [Hernández Orallo et al., 2004]. Para ello se utiliza la siguiente expresión:

$$\Delta w_{ij}(t+1) = (1 - \alpha) \delta_j o_i + \alpha \Delta w_{ij}(t) \quad (6.24)$$

donde α determina la influencia del momento y estando comprendido entre cero y uno.

6.3.8.K-vecinos

k-Vecinos es una técnica supervisada no paramétrica.

En este método se asigna la clase mayoritaria entre los k vecinos más próximos, siendo el valor de k muy importante y a la vez difícil de establecer [Hernández Orallo et al., 2004].

Siendo:

- **n** el número total de patrones.
- **c** el número total de clases.
- **k** el valor elegido como la cantidad de vecinos más cercanos a x.
- **w_i** la clase i, i=1,...c.

- n_i la cantidad total de patrones pertenecientes a la clase i .
- k_i el número de patrones vecinos más cercanos pertenecientes a la clase i .
- $V(x)$ el volumen que delimita los k vecinos más cercanos de x .

En este método se toma como la probabilidad de una clase w_i a priori y luego se calculan las probabilidades a posteriori directamente, para seleccionar la clase con mayor presencia entre los k vecinos que rodean a x , esto es:

$$p(w_i) = \frac{n_i}{n} \quad (6.25)$$

$$p(x|w_i) = \frac{k_i}{n_i \cdot V(x)} \quad (6.26)$$

$$p(x, w_i) = p(x|w_i) \cdot p(w_i) = \frac{k_i}{nV(x)} \quad (6.27)$$

$$p(w_i|x) = \frac{p(x, w_i)}{\sum_{j=1}^c p(x, w_j)} = \frac{k_i}{k} \quad (6.28)$$

Y por lo tanto se selecciona w_j si se cumple:

$$k_j = \max_{i=1, \dots, c} \{k_i(x)\} \quad (6.29)$$

En el algoritmo 6.4 se define el comportamiento de k -vecinos.

Algoritmo 6.5 K-vecinos

Entrada: D = dataset con N casos y caracterizado por n variables predictoras.

x = nuevo caso a clasificar

Para todo objeto ya clasificado (x_i, c_i)

 calcular $d_i = d(x_i, x)$

Fin Para

Ordenar $d_i (i=1, \dots, N)$ en orden ascendente

Quedarnos con los K casos D_x^K ya clasificados más cercanos a x

Asignar a x la clase más frecuente en D_x^K

En este método, para la estimación de la distancia se utilizan dos estrategias conocidas como distancia euclídea y distancia Mahalanobis, las cuales se presentan a continuación en su forma vectorial para una distancia entre dos vectores:

$$D_{euclídea}(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T (\vec{x} - \vec{y})} \quad (6.30)$$

$$D_{Mahalanobis}(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \sum_{-I} (\vec{x} - \vec{y})} \quad (6.31)$$

6.4. Análisis de los resultados

Este apartado tiene como objetivo valorar diferentes modelos como clasificadores. La idea es conseguir un modelo capaz de discriminar las tres clases (concentrado, desconcentrado e inactivo) con el mínimo error posible. Para el entrenamiento de los datos, se ha utilizado el método de validación cruzada con 5 pliegues, en el que 4 de los 5 pliegues se utilizan para entrenar el algoritmo, y el último pliegue se utiliza como set de test.

A continuación se detallarán los resultados obtenidos con cada uno de los métodos anteriormente descritos.

6.4.1. C4.5

En este apartado se van a analizar los resultados obtenidos del modelo basado en C4.5. Para ello se ha seleccionado en WEKA el método J48 y se ha utilizado un factor de confianza de 0.15.

A continuación se muestra la matriz de confusión obtenida del entrenamiento de este algoritmo y los valores AUC/ROC:

Clasificado como ->	Concentrado	Distraído	Inactivo
Concentrado	519	13	138
Distraído	30	57	2
Inactivo	121	5	474

Tabla 6.5: Matriz de confusión del algoritmo C4.5

Concentrado	0,779
Distraído	0,861
Inactivo	0,813
Media	0,799

Tabla 6.6: Valores AUC/ROC obtenidos en C4.5

6.4.2.K-vecinos

Para la ejecución de este método se ha seleccionado el método IBK de WEKA. Puesto que el valor k resulta un valor crítico a la hora de entrenar el modelo, y puesto que es dependiente de la distribución de los datos y por lo tanto del dataset concreto, se ha optado por realizar una serie de ejecuciones del modelo variando el valor de k y midiendo el error cuadrático medio para determinar el valor de k que menos error produzca. El valor de k elegido finalmente ha sido k=11.

Para realizar el cálculo de la distancia, se ha utilizado la distancia euclídea, y no se ha aplicado ningún factor peso sobre la distancia.

A continuación se muestra la matriz de confusión obtenida del entrenamiento de este algoritmo y los valores AUC/ROC:

Clasificado como ->	Concentrado	Distraído	Inactivo
Concentrado	511	15	144
Distraído	36	52	1
Inactivo	123	5	472

Tabla 6.7: Matriz de confusión del algoritmo KNN

Concentrado	0,831
Distraído	0,942
Inactivo	0,861
Media	0,852

Tabla 6.8: Valores AUC/ROC obtenidos en KNN

6.4.3.Redes Neuronales Artificiales

Este método corresponde a MultilayerPercetron en WEKA. En este método se han establecido el parámetro momentum a 0.1 y el ratio de aprendizaje a 0.5, y se ha creado una capa oculta con 6 neuronas:

En la figura 6.3 se muestra la topología de la red neuronal utilizada:

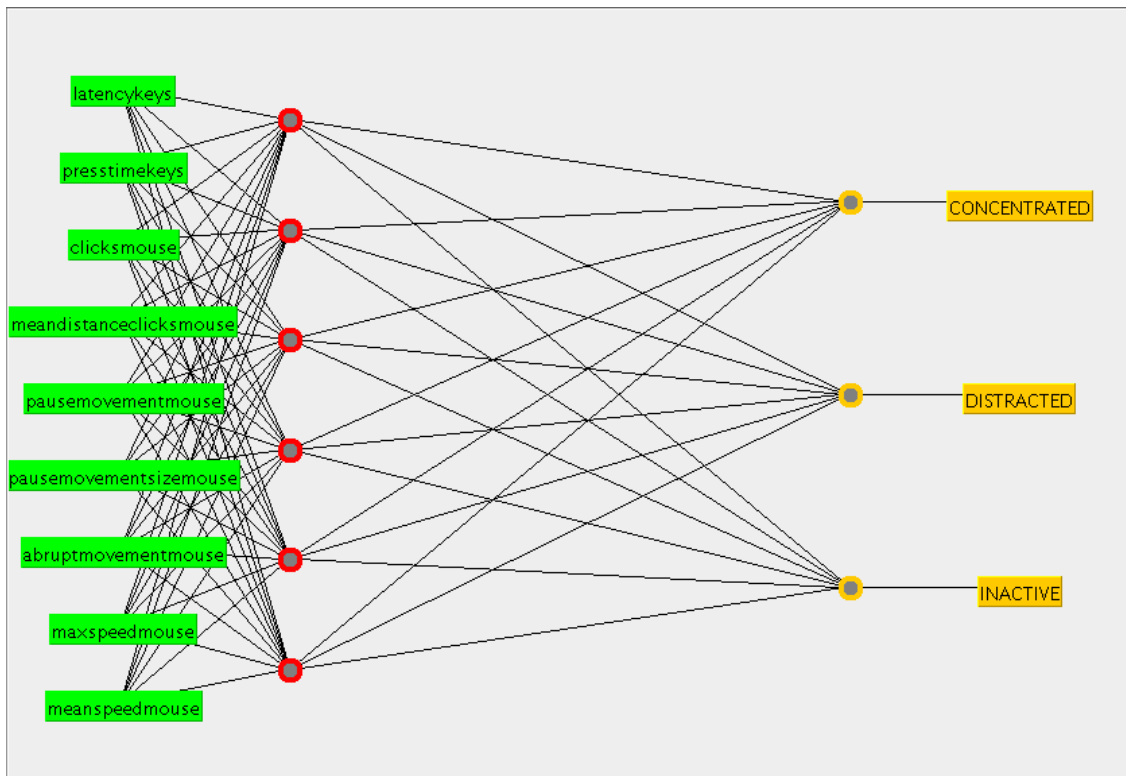


Figura 6.6: Topología de la red neuronal utilizada para la clasificación

A continuación se muestran la matriz de confusión obtenida del entrenamiento de este algoritmo y los valores AUC/ROC:

Clasificado como ->	Concentrado	Distraído	Inactivo
Concentrado	444	24	202
Distraído	23	64	2
Inactivo	57	3	540

Tabla 6.9: Matriz de confusión del algoritmo de redes neuronales

Concentrado	0,818
Distraído	0,883
Inactivo	0,858
Media	0,84

Tabla 6.10: Valores AUC/ROC obtenidos en la red neuronal

6.4.4. Máquinas de Vectores Soporte

Para la ejecución de este método se ha utilizado el método SMO de WEKA. En este método se ha establecido el parámetro de complejidad a 0.5, se ha utilizado RBF como kernel donde se ha establecido gamma a 0.5.

A continuación se muestran la matriz de confusión obtenida del entrenamiento de este algoritmo, y los valores AUC/ROC:

Clasificado como ->	Concentrado	Distraído	Inactivo
Concentrado	488	12	170
Distraído	38	50	1
Inactivo	95	6	499

Tabla 6.11: Matriz de confusión del algoritmo SVM

Concentrado	0,769
Distraído	0,864
Inactivo	0,809
Media	0,793

Tabla 6.12: Valores AUC/ROC obtenidos en SVM

6.4.5. Naïve Bayes

La ejecución de este método en WEKA se realiza mediante la selección de NaiveBayes en el selector del clasificador.

Para la ejecución de este algoritmo se ha utilizado el estimador del kernel.

A continuación se muestra la matriz de confusión obtenida del entrenamiento de este algoritmo, y los valores AUC/ROC:

Clasificado como ->	Concentrado	Distraído	Inactivo
Concentrado	429	43	198
Distraído	17	70	2
Inactivo	67	14	519

Tabla 6.13: Matriz de confusión del algoritmo Naive Bayes

Concentrado	0,815
Distraído	0,946
Inactivo	0,856
Media	0,847

Tabla 6.14: Valores AUC/ROC obtenidos en Naive Bayes

6.4.6. Conclusiones

Observando las matrices de confusión, se comprueba que existe un alto porcentaje de confusión entre el estado “distraído” y el estado “concentrado” en la mayoría de los métodos, excepto en Naïve Bayes, así como entre los estados “concentrado” e “inactivo”.

Sin embargo, aunque el porcentaje de error ha sido en general elevado, disponemos de valores AUC (Area Under the Curve) de ROC bastante buenos (cerca de 0,8 o superiores).

La gráfica ROC es una técnica para visualizar, organizar y seleccionar clasificadores basándose en el rendimiento de éstos. La curva ROC tiene una propiedad interesante, y es que aunque la distribución de los datos cambie, esta curva no cambia, por lo que es independiente de la distribución de los datos, a diferencia de otras mediciones (curva Recall,...).

Para realizar la comparación de esta curva para los diferentes clasificadores se puede reducir a un valor escalar (entre 0 y 1), calculando el área existente debajo de ella (lo que llamamos AUC), que es el equivalente a la probabilidad de que el clasificador puntúe más alto una instancia aleatoria positiva que una instancia aleatoria negativa.

Basándonos en la media del AUC/ROC podemos decir que por orden de mejores resultados tenemos:

1. K-vecinos: 0,852
2. Naïve Bayes: 0,847
3. Redes Neuronales Artificiales: 0,84
4. C4.5: 0,799
5. Máquinas de Vectores Soporte: 0,793

En este sentido K-vecinos presenta mejores resultados, sin embargo por un lado este método ha sido entrenado con un valor "k" basado en el set de datos (el valor k que menor error presentaba) lo que hace que sin tener un set de datos de mayor tamaño para poder comprobarlo, quepa la posibilidad de haber incurrido en un sobreajuste. Por otro lado, k-vecinos es un método retardado, y estos métodos presentan el inconveniente de que la predicción puede ser lenta y depende del conjunto de ejemplos.

Cabe destacar que el uso de Máquinas de Vectores Soporte ha sido inédito en este tipo de aplicación, y a pesar de que ha sido el método que peores resultados ha ofrecido, se ha aproximado bastante a los mejores resultados, lo que lo habilita para posibles futuros estudios dentro de este tipo de aplicación.

Finalmente, para la generación del modelo se utilizará el método Naïve Bayes con el estimador del kernel, ya que ha presentado un buen comportamiento con variables continuas, así como buenos resultados en el AUC/ROC, y su parametrización es mínima e independiente del set de datos.

CAPÍTULO 7

DESARROLLO DEL MODELO DE INTERVENCIÓN PEDAGÓGICA

En este capítulo se detalla el desarrollo del modelo de intervención pedagógica para la práctica con Scratch.

7. Desarrollo del modelo de intervención pedagógica para la práctica con Scratch

La arquitectura ASIART comprende un modelo de intervención pedagógica genérico, encargado de generar el comportamiento del robot según el modelo del estudiante y su estado afectivo.

La implementación de este modelo es dependiente del dominio y software educativo, por lo que para cada aplicación que implemente la arquitectura ASIART, el modelo de intervención pedagógica puede ser diferente.

Hasta el momento hemos generado un modelo de intervención pedagógica básico que contiene las reglas particulares de Scratch, y en el que tenemos intención de profundizar en el futuro. Dicho módulo se ha desarrollado en base una serie de directrices presentes en [Hernando, 2015] que dependen de la motivación y competencia de los alumnos. Dichas directrices son:

	<i>Menor motivación</i>	<i>Mayor motivación</i>
<i>Mayor competencia</i>	Tutor de la inspiración	Tutor Coach
	<ul style="list-style-type: none"> - Se busca despertar el interés y la motivación por la tarea. - Se centra en más ejemplos y casos prácticos para representar el contenido en la vida cotidiana o en otros proyectos y áreas de la escuela. - Se acentúan las recompensas en periodos cortos de tiempo. - Se pregunta acerca de la falta de vinculación con el contenido y se exploran posibles soluciones. - Se usan acertijos y juegos relacionados con el contenido. 	<ul style="list-style-type: none"> - Se deja libertad para experimentar y cometer errores. - Se presentan contenidos o problemas de mayor dificultad que animen a nuevos retos. - Se mantiene el interés y se pregunta acerca de la vinculación con el contenido y por los elementos más satisfactorios o por las partes favoritas. - Se anima a que sean tutores de otros compañeros. - Se fomentan los momentos reflexivos acerca de la naturaleza del contenido y su utilidad.
<i>Menor competencia</i>	Tutor directivo	Tutor Guía
	<ul style="list-style-type: none"> - Se marcan metas concretas y sencillas en periodos cortos de respuesta. - Se organizan secuencias muy dirigidas con recompensa constante. - Se mantiene un contacto y un seguimiento cercano. - Se representa el progreso de forma gráfica. - Se buscan ejemplos muy concretos del uso del contenido en la vida cotidiana. - Se pueden repetir los mismos ejercicios en más de una ocasión. - Se trata de establecer relaciones con las estrategias en otras áreas en las que haya una mejor competencia o una mayor motivación. 	<ul style="list-style-type: none"> - Se visualiza el futuro cuando las metas de aprendizaje se hayan alcanzado. - Se trabaja en un modo más rápido, centrado en el aumento de la dificultad de la tarea. - Se ofrecen más tareas y actividades para repetir las y hacerlas por su cuenta. - Se incide mucho en las estrategias cognitivas para resolver un problema o contenido y en su representación gráfica. - Se fortalece la motivación con metas y recompensas a medio y largo plazo.

Tabla 7.1: Tipo de tutor según la competencia y motivación del aula (fuente: [Hernando, 2015]).

Para desarrollar un modelo de intervención pedagógica concreto, la plataforma ASIART comprende una herramienta parametrizable para generar las reglas de las intervenciones pedagógicas, así como la definición de las intervenciones. Un ejemplo de tales reglas sería el siguiente:

IF (competente = TRUE Y motivado = TRUE) THEN Intervención_1;

IF(competente = TRUE Y motivado = TRUE) THEN Intervención_2;

Donde:

Intervención_1: se busca despertar el interés y la motivación por la tarea

{

Diálogo = (aquí se proporciona una sugerencia de diálogos que habrán de adaptarse a cada software educativo y escenario de aprendizaje concretos, por ejemplo “¡He visto que lo estás haciendo genial! ¡Sigue así y llegarás a ser tan sabio como Splinter!”).

Tono = (aquí se proporciona una sugerencia de tonos adecuados, por ejemplo "tono alegre, tono orgulloso)

Gestos = (aquí se proporciona una sugerencia de gestos adecuados, por ejemplo: "levantar un brazo").

}

Intervención_2: se centra en más ejemplos y casos prácticos

{

Diálogo = (aquí se proporciona una sugerencia de diálogos que habrán de adaptarse a cada caso concreto, por ejemplo “Voy a ponerte un ejemplo...”)

Tono = (aquí se proporciona una sugerencia de tonos adecuados, por ejemplo "tono amable, tono sugerente)

Gestos =(aquí se proporciona una sugerencia de gestos adecuados, por ejemplo: "llevarse la mano a la cabeza")

}

De este modo, las grabaciones de las experiencias servirían sobre todo para identificar los escenarios que requieren intervención educativa para un software educativo concreto. Para implementar el correspondiente modelo de intervención pedagógica, el implementador tendría simplemente que asociar cada escenario con un conjunto de reglas (cada una para un valor de competencia y motivación) y editar las reglas para adaptarlas a cada caso. Las reglas se traducirían de forma automática a especificaciones de un avatar afectivo y luego a acciones del robot, sin necesidad que el implementador tenga que conocer los correspondientes lenguajes de especificación.

CAPÍTULO 8

IMPLEMENTACIÓN DE ASIART

En este capítulo se detallan las características de la plataforma de desarrollo y su funcionamiento.

8. Plataforma de desarrollo de robots tutores afectivos en el marco de la arquitectura ASIART

En este capítulo se describe el proceso de desarrollo, en el marco de la arquitectura ASIART, de un robot tutor afectivo para la tutorización de alumnos que practican con un software educativo.

Como ya se ha visto en el capítulo 3 y tal como se muestra en la figura 8.1 la arquitectura se divide principalmente en tres partes: cliente, servidor web y robot. En los siguientes subapartados se detallarán el proceso de desarrollo y funcionamiento de cada uno de los componentes de la implementación de la arquitectura ASIART.

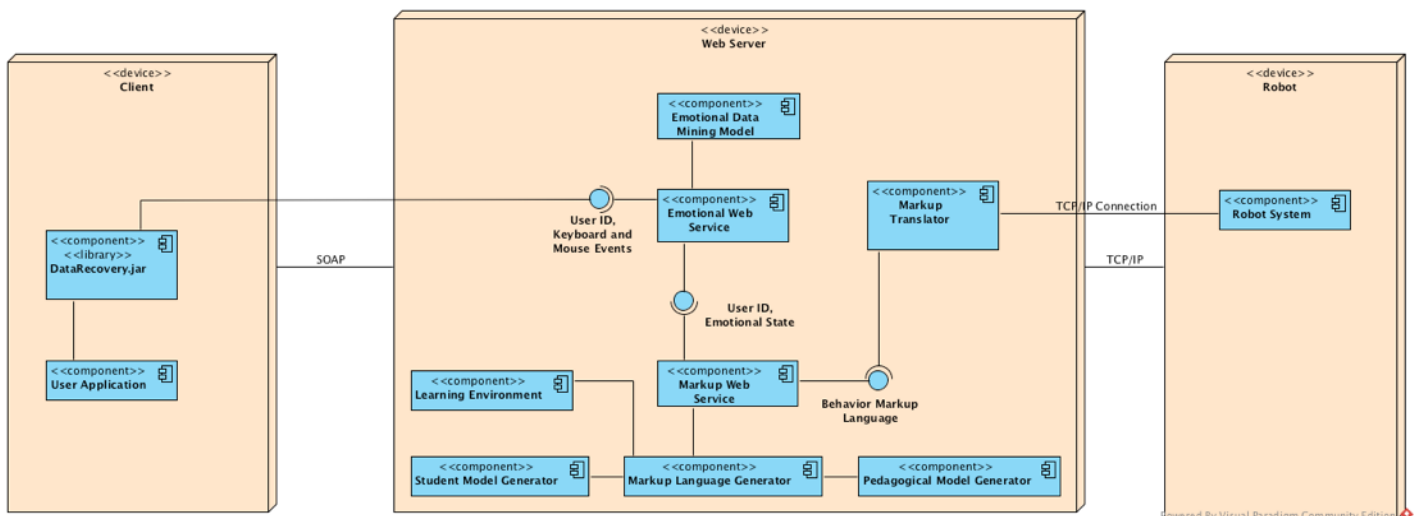


Figura 8.1: Diagrama de despliegue de la arquitectura ASIART

8.1. Cliente (Data Recovery)

La parte cliente es la que se encarga de recibir la información del usuario, tanto su identificador, como las interacciones del teclado y el ratón.

Para las interacciones, se ha desarrollado una API en JAVA que puede ser llamada desde una aplicación educativa, o incluso, en caso de ser necesario, esta API dispone de una interfaz gráfica propia (IHM), desde la cual se puede indicar el ID del usuario y obtener las interacciones.

Esta interfaz gráfica se puede ver en las figuras 8.2 y 8.3.

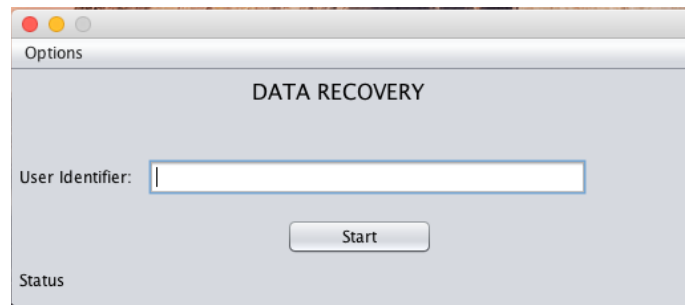


Figura 8.2: Interfaz principal de DataRecovery

En la pantalla principal de la interfaz (figura 8.2) sólo es necesario introducir el identificador del usuario y apretar al botón de iniciar (Start). Con esto, el sistema comenzará el proceso de log de interacciones, realizará su procesado y las enviará por web service.

Esta pantalla también dispone de una pantalla de configuración (figura 8.3).

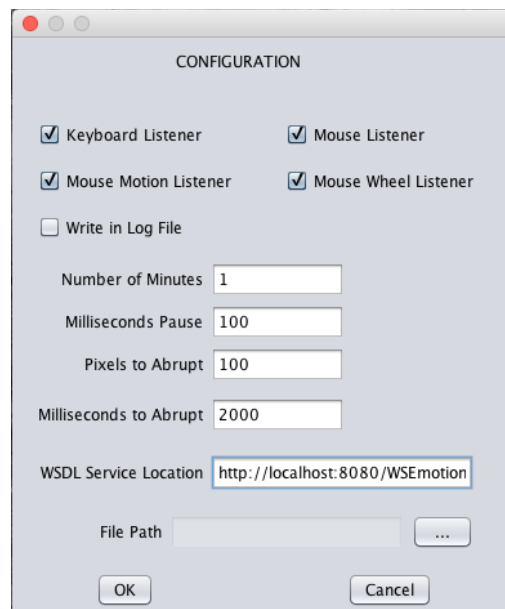


Figura 8.3: Interfaz de configuración de DataRecovery

La API dispone de una clase principal llamada "DataRecovery" con dos métodos principales "Start" y "Stop". Esta clase, en el constructor recibe como parámetro la clase de configuración.

La clase principal y la clase de configuración se pueden ver en el diagrama de clases representado en la figura 8.4.

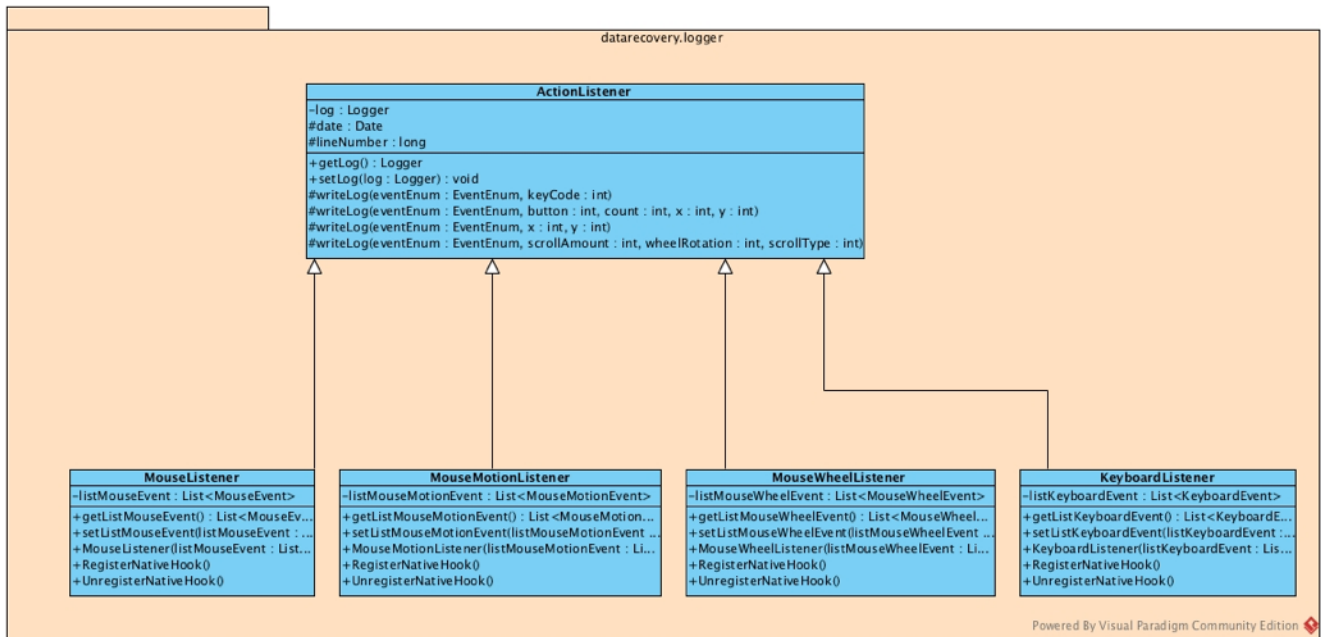


Figura 8.4: Diagrama de clases de las clases principales de DataRecovery

Cuando se llama al proceso “Start” se inicia la suscripción a una serie de “Listeners” de teclado y ratón definidos en la figura 8.5.

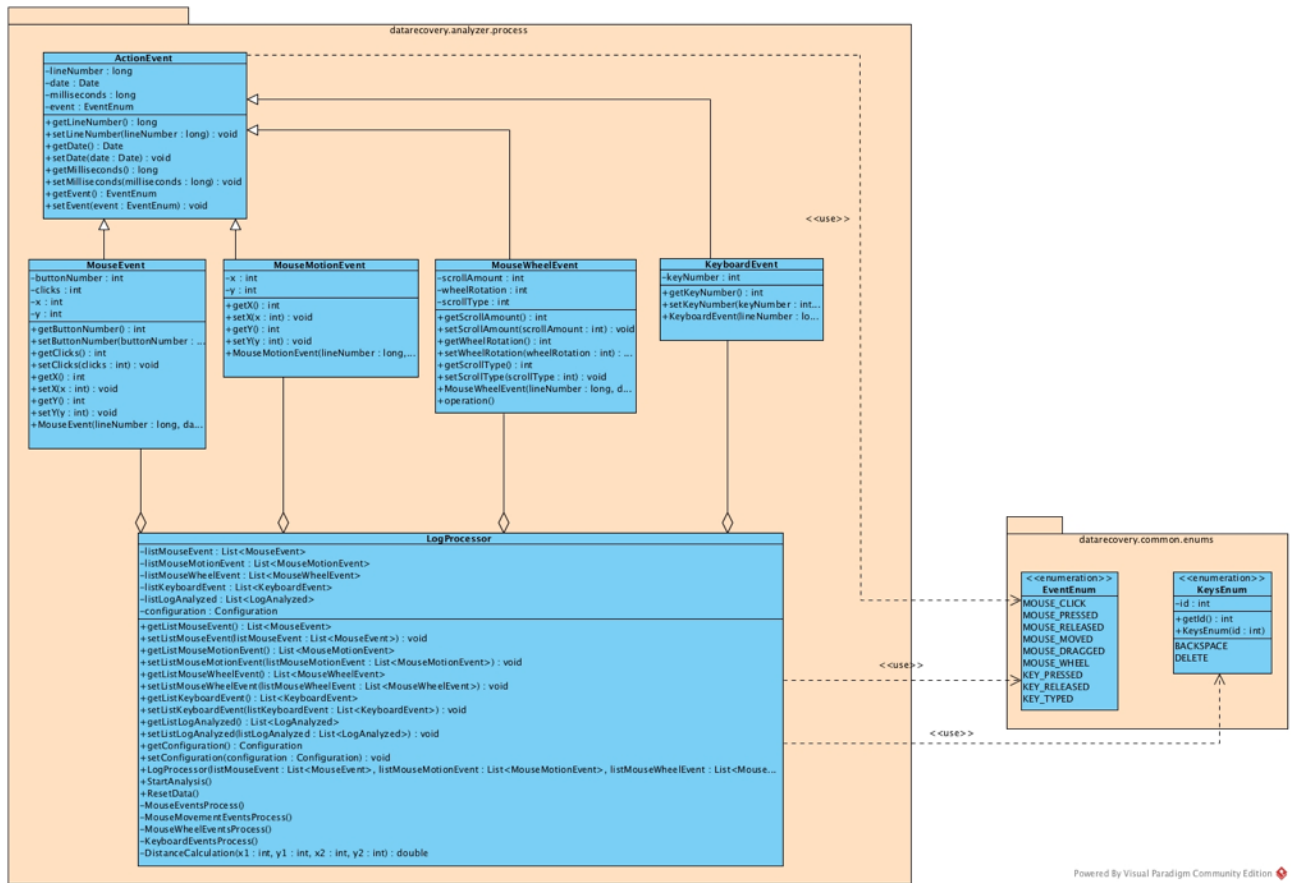


Figura 8.5: Diagrama de clases de los escuchadores de eventos de teclado y ratón de DataRecovery

Estos eventos se van almacenando y periódicamente (según el tiempo que se le haya indicado en la configuración) se procesa la información y se envía al web service emocional (encargado de clasificar el estado emocional del alumno).

Los diagramas correspondientes al procesamiento y al cliente web service son las figuras 8.6 y 8.7.

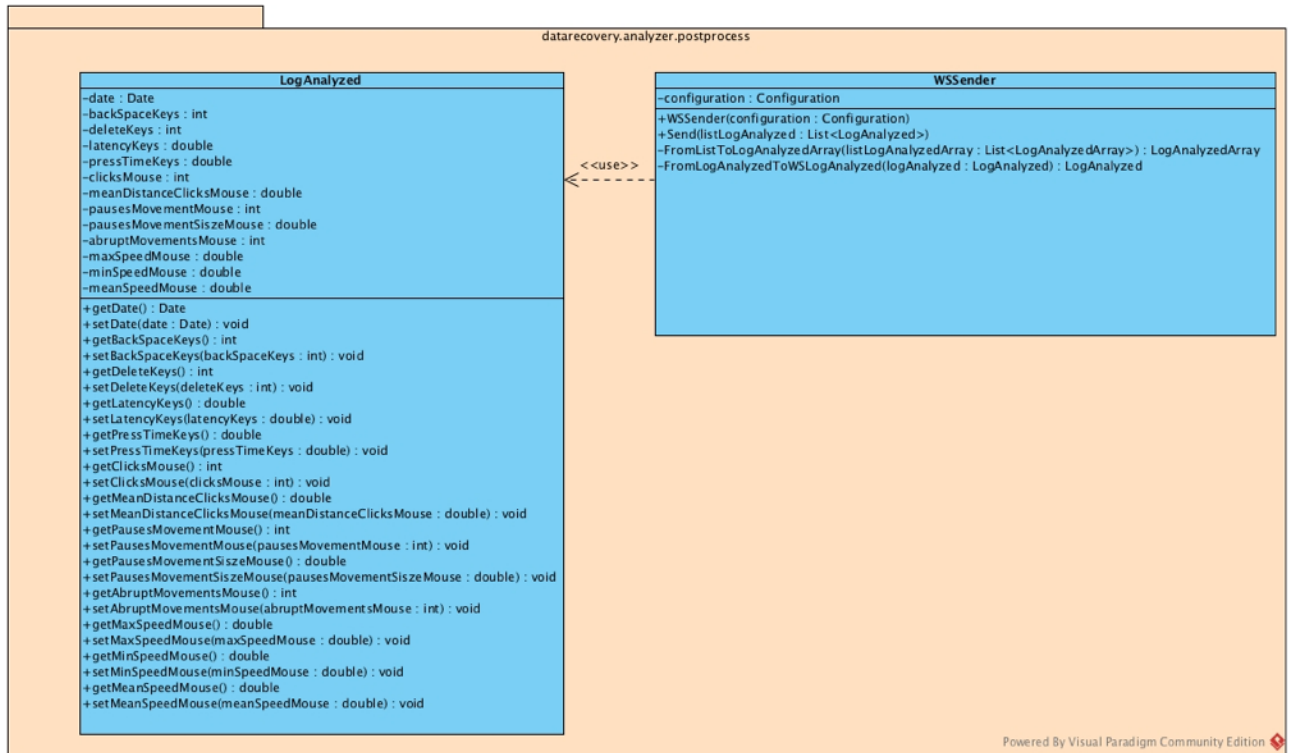


Figura 8.6: Diagrama de clases del procesador de logs de DataRecovery

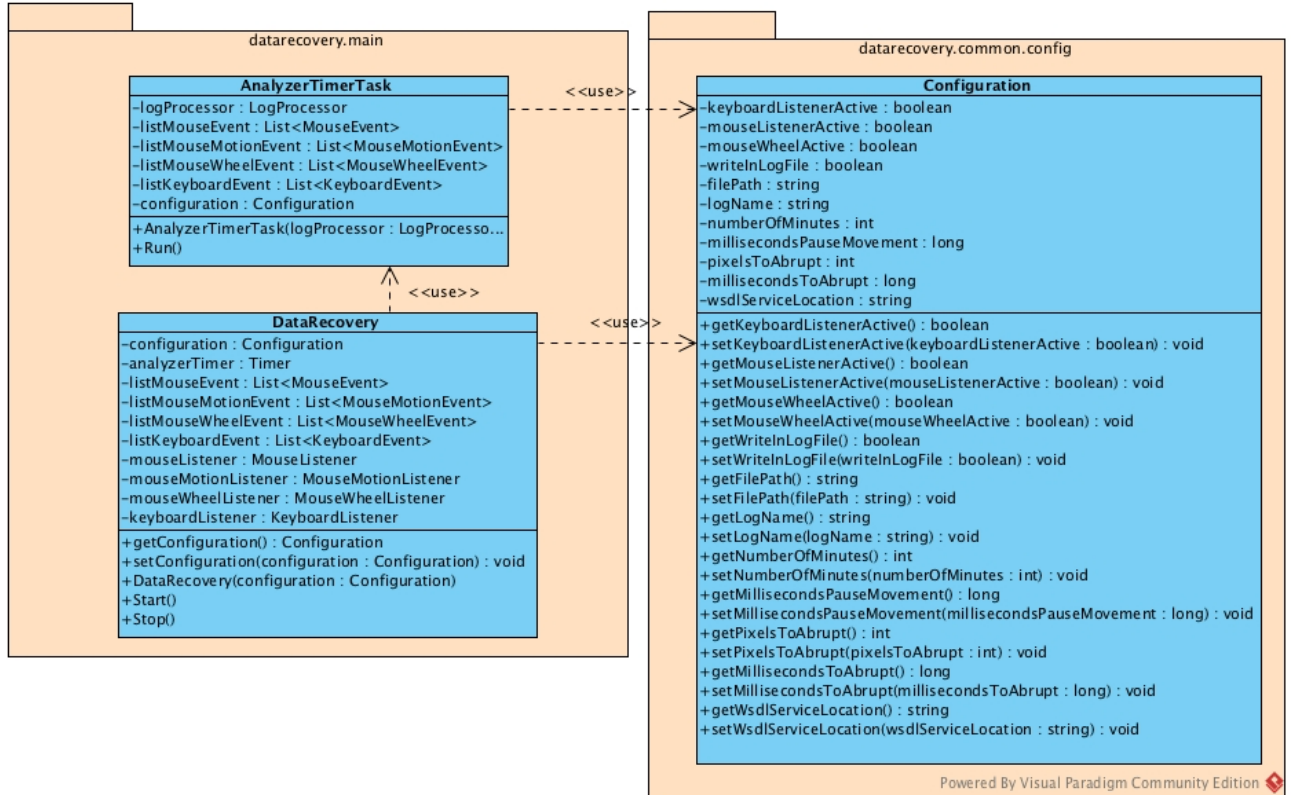


Figura 8.7: Diagrama de clases del post procesador de logs de DataRecovery

A modo ilustrativo se puede ver un esquema del comportamiento en el diagrama de secuencia que se muestra en la figura 8.8.

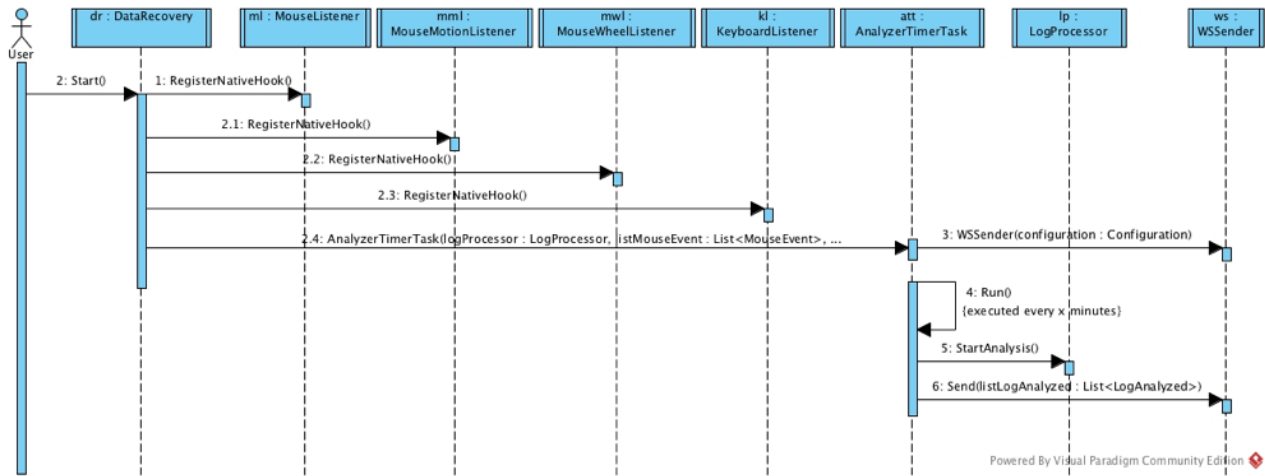


Figura 8.8: Diagrama de secuencia de DataRecovery desde la recogida de datos hasta el envío por servicio web

8.2.Web Server

8.2.1.Emotional Web Service

Este componente trata de un servicio web SOAP que recibe las interacciones de teclado y ratón de la API DataRecovery, así como el identificador del usuario, de manera que posteriormente se pueda extraer el modelo del estudiante.

Este servicio web, además, representa la única interfaz de comunicación entre el cliente y el servidor web, facilitando así el diseño de la arquitectura.

En la figura 8.9 se puede observar el diagrama de clases del servicio web emocional.

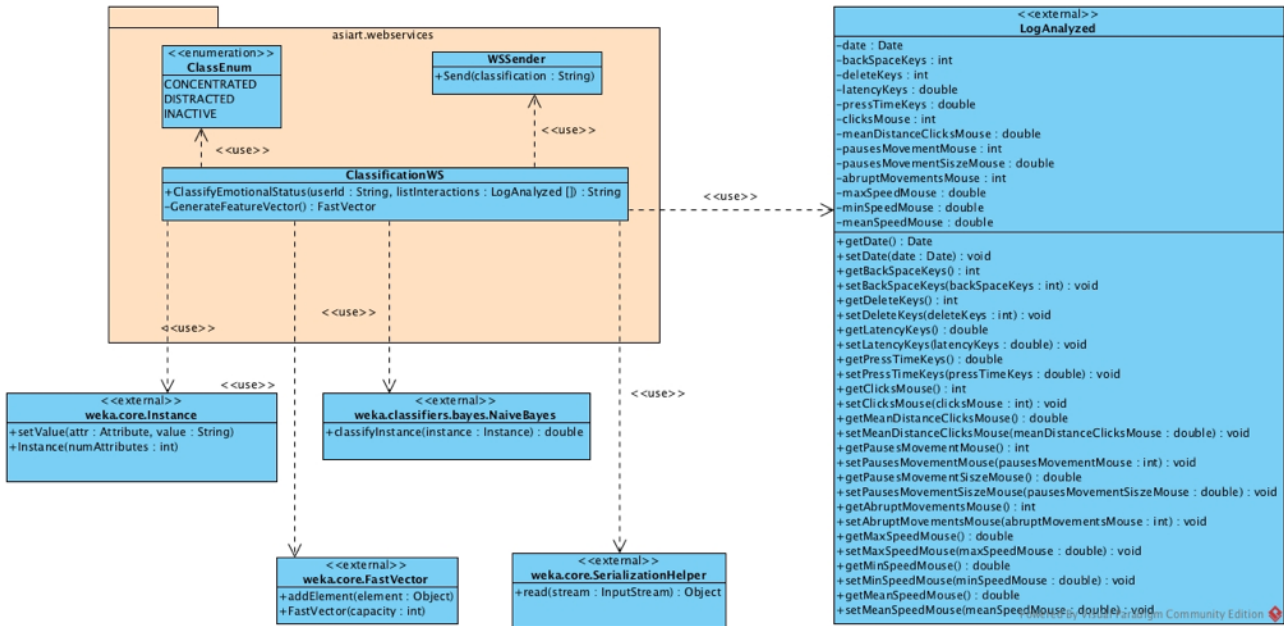


Figura 8.9: Diagrama de clases del servicio web emocional

La función a la que se llama a través de SOAP es “ClassifyEmotionaStatus” de la clase “ClassificationWS”.

Esta función, cuando recibe la llamada con el identificador del usuario y las interacciones del teclado y el ratón, instancia el modelo “NaiveBayes” entrenado anteriormente y a través de un servicio web envía la información de identificación del usuario, así como del estado emocional determinado por el modelo instanciado.

Este estado emocional también será devuelto como resultado de la llamada del web service.

Para mejorar la comprensión del funcionamiento de las llamadas, se ha incluido en la figura 8.10 el diagrama de secuencia del servicio web emocional.

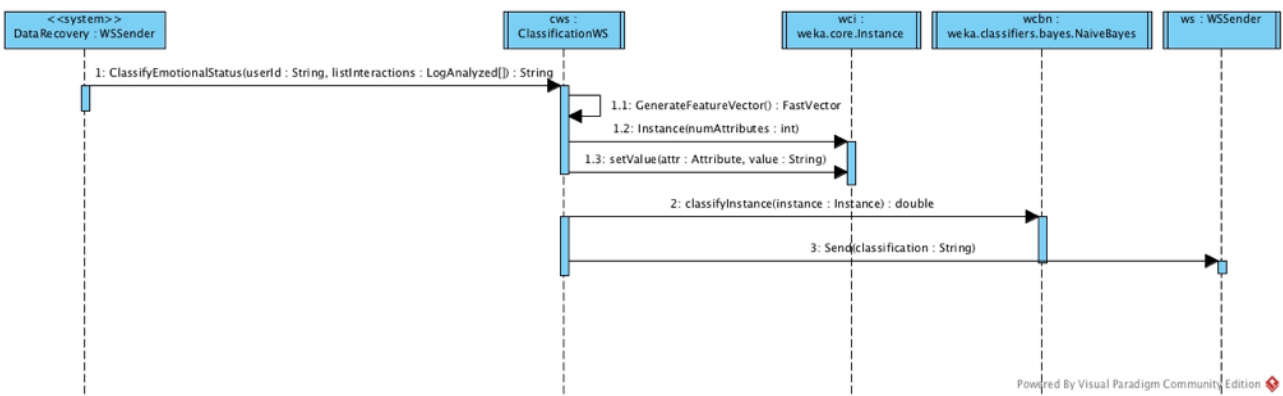


Figura 8.10: Diagrama de secuencia del servicio web emocional

8.2.2. Markup Web Service

Este componente se encarga de generar las interacciones entre el robot y el alumno, en base al perfil del alumno y a su estado emocional.

Para ello, consultará por un lado un generador del modelo de estudiante, en base al identificador del estudiante, y junto a ese modelo y a su estado emocional se realizará una consulta a un generador de modelo pedagógico de donde se obtendrán las acciones que desempeñará el robot bajo el lenguaje de marcado BML.

Finalmente se enviarán las acciones bajo BML a un intérprete que traducirá dichas acciones al robot en cuestión.

En las figuras 8.11 y 8.12 están representados respectivamente los diagramas de clases y de secuencia de este servicio web.

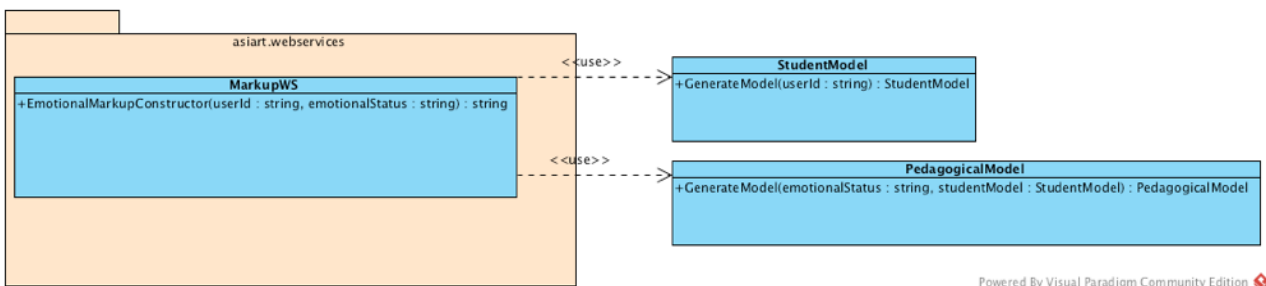


Figura 8.11: Diagrama de clases del servicio web del lenguaje de etiquetado

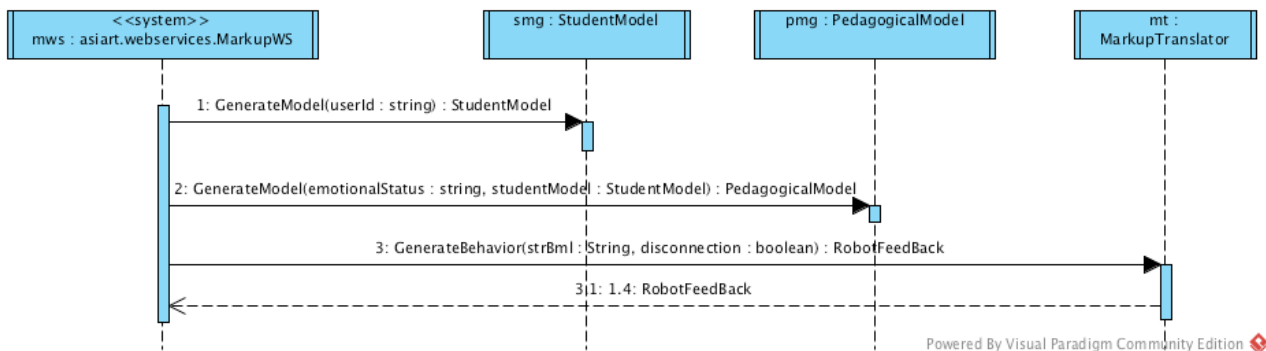


Figura 8.12: Diagrama de secuencia del servicio web del lenguaje de etiquetado y los generadores de modelos de intervención pedagógica y de estudiante

8.2.3. Markup Translator

Este componente es una librería utilizada por el “Markup Web Service” y se encarga de traducir la información que llega en formato BML a las interacciones del robot. Este componente es específico de cada robot.

9. Implementación de ASIART con SCRATCH y NAO

En este apartado se describe el desarrollo llevado a cabo para implementar ASIART con SCRATCH como software educativo y con NAO como robot tutor. En este proyecto se ha realizado una implementación con muchas simplificaciones.

Puesto que “DataRecovery.jar” ya recupera la información de las interacciones de teclado y ratón, así como el identificador del usuario, no ha sido necesario realizar ninguna modificación en SCRATCH.

En la figura 9.1 se puede observar un esquema de la arquitectura con la implementación de SCRATCH y NAO.

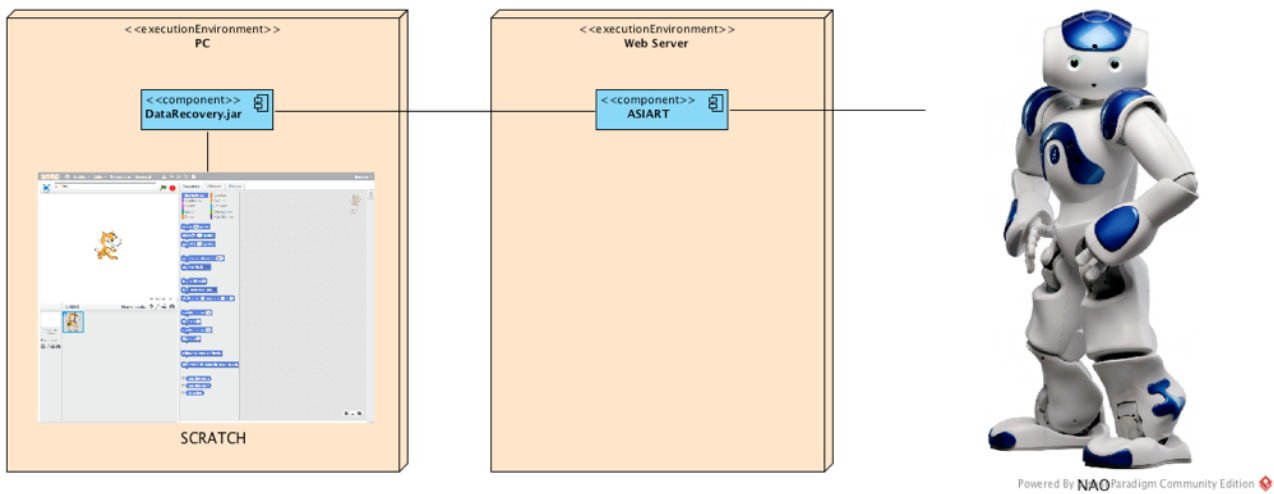


Figura 9.1: Arquitectura ASIART con la implementación de Scratch y NAO.

Para implementar la arquitectura ASIART en cualquier robot, el único componente que hay que adaptar es el “Markup Translator”, de manera que la interpretación de BML se traduzca a acciones propias de cada robot.

Para traducir el lenguaje BML a las acciones de NAO, se ha creado una serie de clases que sirven para transformar el texto BML a objetos. Estas clases están representadas en la figura 9.2.

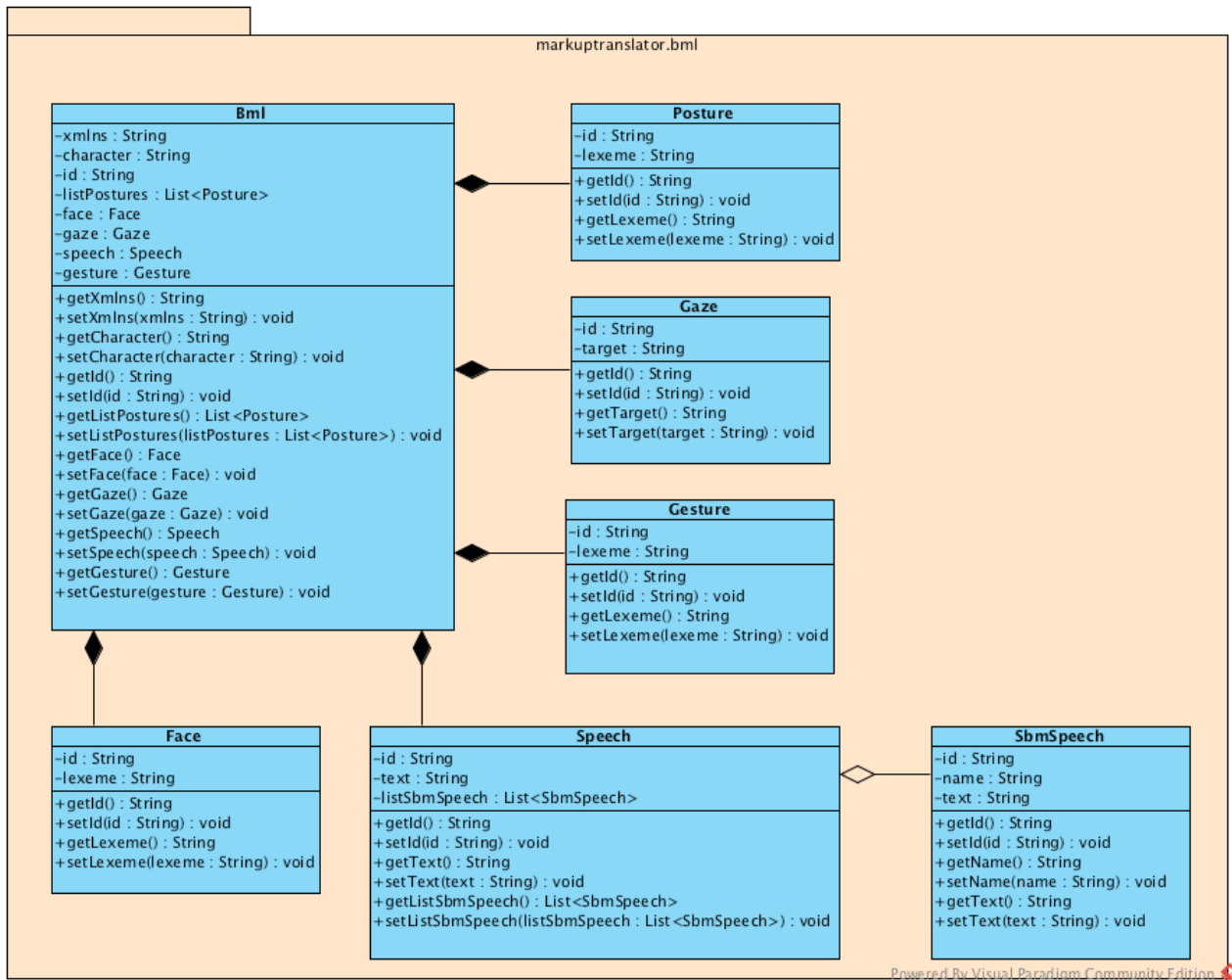


Figura 9.2: Diagrama de clases para la transformación de BML

Una vez que se ha transformado el texto BML a objetos BML, se realiza una traducción de estos objetos BML a acciones del robot. Para esta transformación se han creado las clases ilustradas en la figura 9.3.

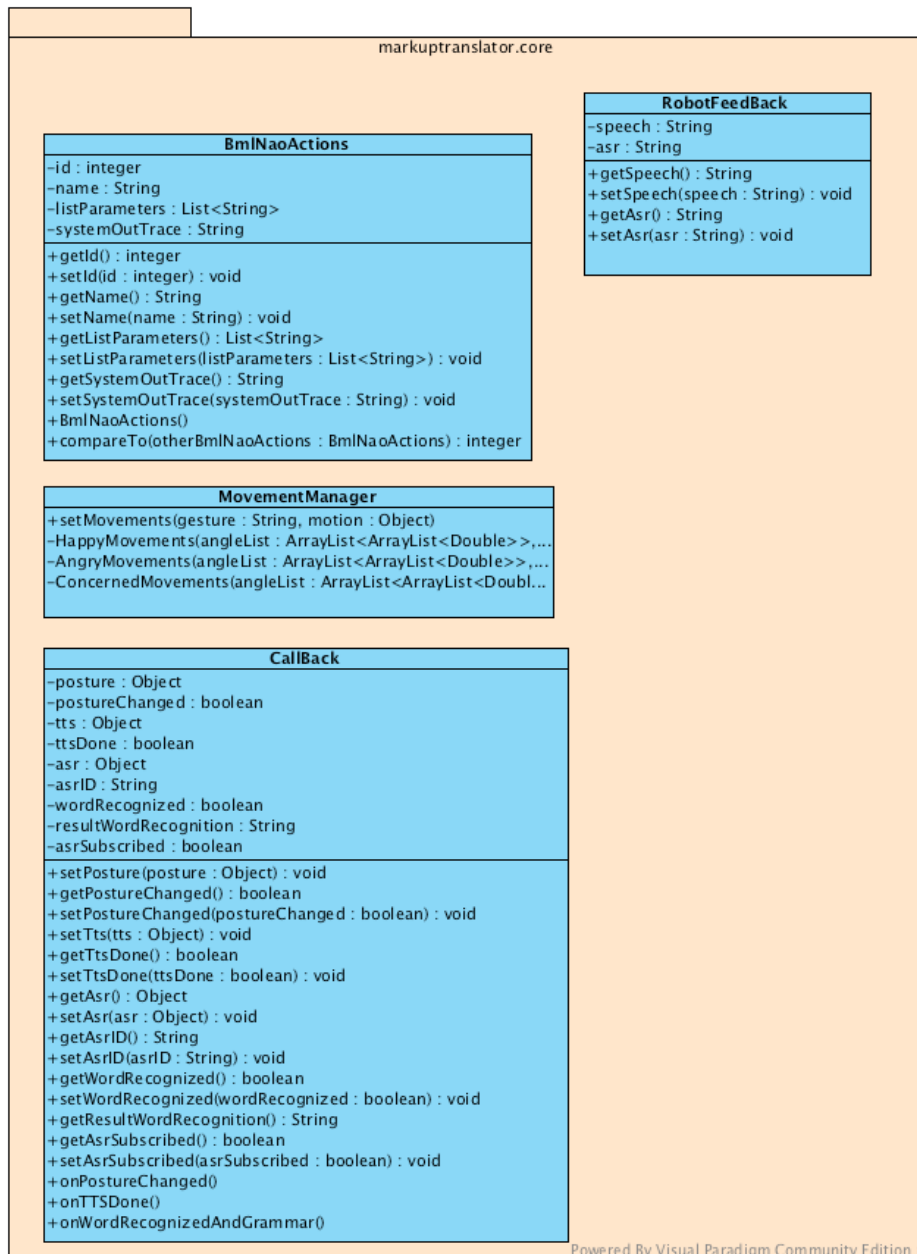


Figura 9.3: Diagrama de clases para la traducción de BML a acciones de NAO

NAO presenta una API que permite el desarrollo en varios lenguajes de programación diferentes tales como Python, C++ y Java. Puesto que la arquitectura ha sido desarrollada en Java, se ha seguido utilizando este lenguaje para las llamadas a las funciones del robot.

Las interacciones de NAO permiten movimientos de motor (brazos, manos, cabeza, etc), color de leds (cabeza, orejas y ojos), guiños (encendido y apagado de los leds de los ojos), así como TTS (Text To Speech) y STT (Speech To Text) entre otras muchas funciones.

Una de las partes más importantes de este proyecto, además de la detección de las emociones mediante las interacciones de teclado y de ratón, es el diálogo con el alumno.

Para generar un diálogo fluido y que sea capaz de ayudar al alumno, en cada interacción se consulta el generador del modelo pedagógico. Para ello, cuando el robot detecta la respuesta del alumno (mediante STT, que se recibe en el módulo del traductor del lenguaje de marcado), ésta respuesta se devuelve al servicio web del lenguaje de etiquetado a través de un objeto de tipo “RobotFeedBack”. El servicio web del lenguaje de etiquetado volverá a consultar al generador del modelo pedagógico creando así los nuevos comportamientos que debe adoptar el robot.

Este módulo específico, además de lo mencionado, activa en NAO el seguimiento con la mirada al alumno al que se está dirigiendo, de forma que el alumno sepa en todo momento que el robot se está dirigiendo a él cuando le habla.

El diagrama de clases de la clase principal de la librería de traducción del lenguaje BML a NAO está ilustrado en la figura 9.4.

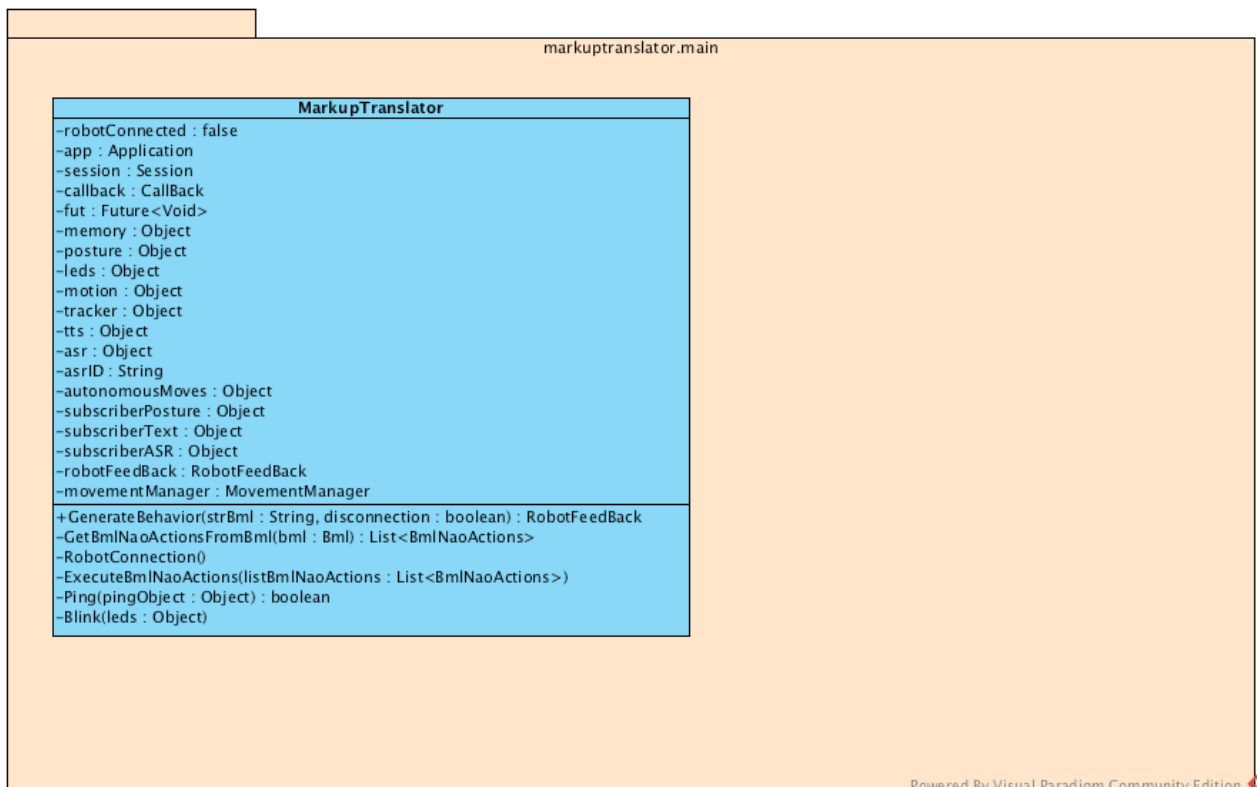


Figura 9.4: Diagrama de clases de la clase principal de la librería de traducción de BML a NAO.

En la figura 9.5 se puede observar el diagrama de secuencia de las interacciones entre el servicio web del lenguaje de marcado y la librería de traducción de BML a instrucciones de NAO.

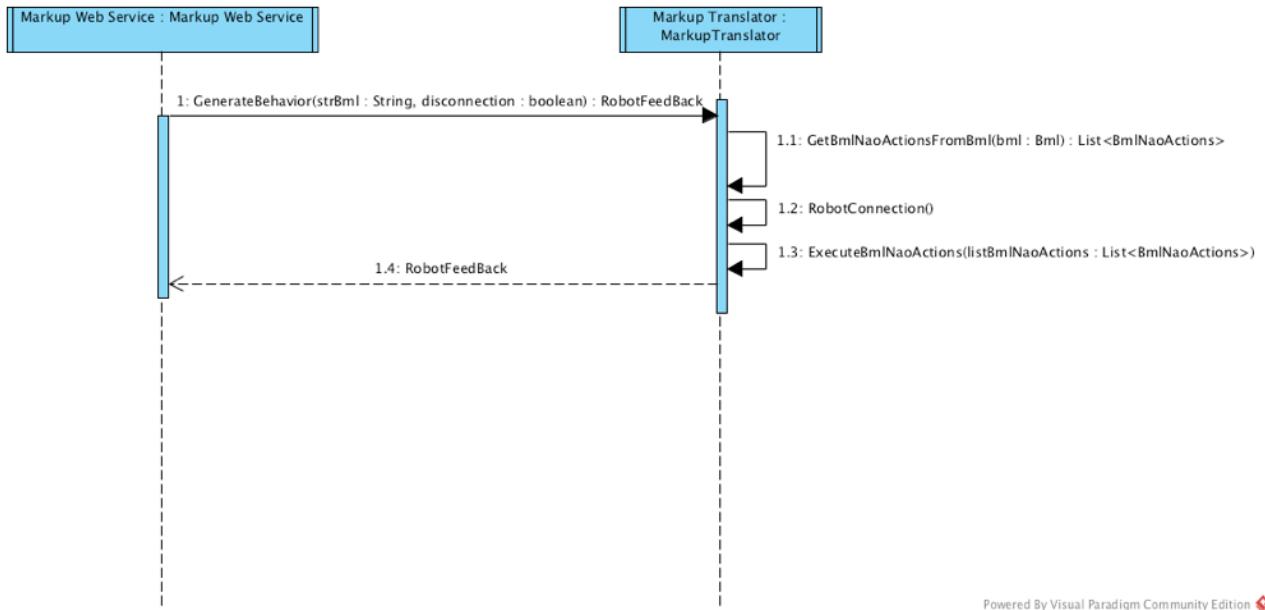


Figura 9.5: Diagrama de secuencia del servicio web del lenguaje de etiquetado y la librería de traducción de BML a NAO.

CAPÍTULO 9

EXPERIENCIA DE TUTORIZACIÓN

En este capítulo se describen la preparación y la realización de una experiencia para la validación preliminar del robot tutor para la enseñanza con Scratch.

10. Experiencia de tutorización

La arquitectura ASIART descrita en el capítulo 3 ha sido evaluada en una experiencia conducida con alumnos de primaria. Para llevar a cabo esta experiencia, se ha implementado la arquitectura tal como se describe en el capítulo 7, se han diseñado unos ejercicios para Scratch y se han escogido dos alumnos de primaria pertenecientes a modelos de estudiante diferentes. La experiencia ha sido registrada por una cámara fija detrás del alumno a una distancia en la que se pudiera entender los diálogos entre el alumno y NAO.

La experiencia ha consistido en la realización de los ejercicios propuestos por separado, de forma que se pueda apreciar la reacción de NAO frente a los distintos estados cognitivo-afectivos y a los diferentes modelos de estudiante.

Uno de los modelos de estudiante utilizado en la experiencia ha sido el de un alumno con alta motivación y alta competencia. En esta experiencia, a pesar de que el alumno ha estado en algún momento experimentando con elementos que no pertenecían completamente al ejercicio propuesto, el robot ha animado al alumno a continuar realizando las tareas que estaba realizando hasta el momento.

El otro modelo de estudiante utilizado en la experiencia ha sido el de un alumno con baja motivación y alta competencia. En esta experiencia, cuando el alumno estaba realizando tareas diferentes de aquellas encomendadas por el profesor, el robot intervenía para intentar motivar el interés por las tareas y guiar al alumno a la hora de realizar los ejercicios.

Finalmente las grabaciones realizadas en la experiencia han sido archivadas para su posterior análisis por un experto.

En las figuras 10.1 y 10.2 se muestran dos fotogramas de la experiencia de tutorización realizada.

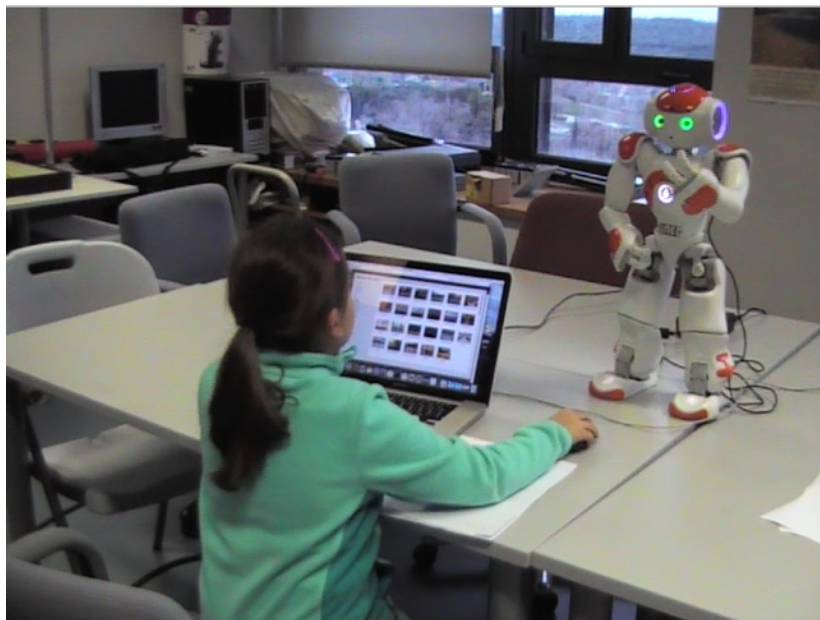


Figura 10.1: Fotograma de la experiencia de tutorización

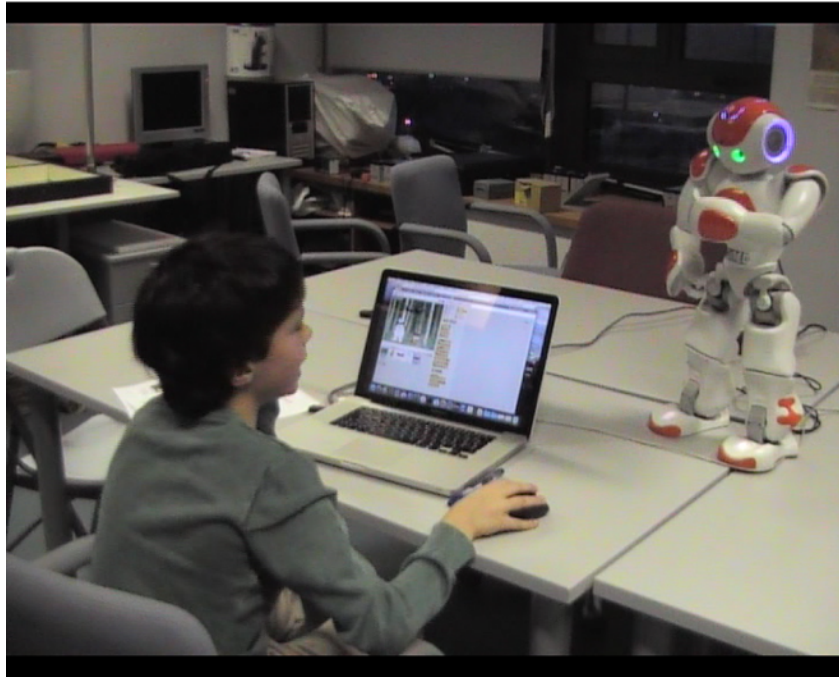


Figura 10.2: Fotograma de la experiencia de tutorización

CAPÍTULO 10

EVALUACIÓN DE ASIART

En este capítulo se exponen los resultados obtenidos de la encuesta realizada a los alumnos, así como una discusión de los mismos.

11. Evaluación de ASIART

La experiencia descrita en el capítulo 8 ha sido evaluada por los dos alumnos que participaron en la experiencia. Para ello se ha creado una encuesta basada en [Chuttur, 2009] [Liu et al., 2010] [Regueiro, 2009] [Selim, 2003], dividida en 5 secciones cuyas preguntas tienen asociada una escala Likert de 5 grados siendo (1) “totalmente desacuerdo” y (5) “totalmente de acuerdo”. A continuación de la encuesta propiamente dicha se propone otra sección con 4 cuestiones de respuesta libre.

A continuación se muestra en diferentes tablas cada una de las secciones con las preguntas y respuestas de cada uno de los alumnos, siendo “Alumno 1” el alumno con alta motivación y alta competencia, y “Alumno 2” el alumno con baja motivación y alta competencia:

Sección 1: Elección del medio / aspectos técnicos

Valoración sobre la elección de un robot (y del robot NAO, en particular) para la tutorización afectiva de niños de primaria que practican con el software educativo Scratch.

	Alumno 1	Alumno 2
Me gusta que un robot tutorice mi aprendizaje	5	3
Prefiero un robot tutor a un agente virtual para tutorizar mi aprendizaje	3	5
Prefiero un robot tutor a un profesor humano	1	2
Prefiero tener un robot tutor antes que trabajar solo	4	5
Me gusta el robot NAO	5	4

Tabla 9.1: Respuestas de los alumnos a la sección 1 de la evaluación

Sección 2: Actuación del robot

Valoración de cómo opera el robot en la práctica.

	Alumno 1	Alumno 2
El robot identificaba las situaciones que me surgían en la práctica con Scratch	1	1
Entendía bien sus mensajes	1	2
Los mensajes eran adecuados a las circunstancias	1	2
El robot entendía bien mis respuestas	1	2

Tabla 9.2: Respuestas de los alumnos a la sección 2 de la evaluación

Sección 3: Percepción de utilidad

Valoración del efecto que el uso del robot tutor tiene en el aprendizaje.

	Alumno 1	Alumno 2
Me ha ayudado con los problemas que me han surgido	1	2
Me ha ayudado a aprender a usar Scratch	1	1
He aprendido más que con un profesor	3	1
Me encuentro más relajado trabajando que cuando me supervisa un profesor	5	5
He aprendido más que cuando trabajo en solitario	3	3
Estaba más motivado que cuando trabajo en solitario	1	2
Me he divertido hablando con él	3	3

Tabla 9.3: Respuestas de los alumnos a la sección 3 de la evaluación

Sección 4: Percepción de facilidad de uso

Esta sección muestra la usabilidad de la herramienta.

	Alumno 1	Alumno 2
Me resultaba sencillo comunicarme con él	1	2

Tabla 9.4: Respuestas de los alumnos en la sección 4 de la evaluación

Sección 5: Intención de uso

Esta sección muestra el grado final de aceptación de la herramienta.

	Alumno 1	Alumno 2
Me gustaría volver a participar en otra experiencia de este estilo	5	4
Me gustaría tener robots tutores en el cole en las clases con ordenadores	5	5
He contado a mis compañeros mi experiencia	1	1
He contado a mis profesores mi experiencia	1	1

Tabla 9.5: Respuestas de los alumnos en la sección 5 de la evaluación

Sección con respuestas libres:

A continuación se muestran las cuatro preguntas de respuesta libre con sus correspondientes respuestas:

- Situación en que no te ha podido ayudar:
 - Alumno 1: En todas, no me ha ayudado.
 - Alumno 2: En todas, no me ha ayudado.

- En qué te ha sido más útil:
 - Alumno 1: En darme ánimo.
 - Alumno 2: Me ha motivado.

- Qué te ha gustado más de la experiencia:
 - Alumno 1: Hacer Scratch.
 - Alumno 2: Cuando intentaba ser gracioso.

- Qué te ha gustado menos:
 - Alumno 1: Que el robot no me ayudara más.
 - Alumno 2: Cuando respondía cosas que no venían a cuento.

Conclusiones:

Los resultados obtenidos de la evaluación del prototipo implementado de la arquitectura no han sido muy buenos, debido a factores como que el prototipo era muy básico y que se ha contado sólo con dos niños para la experiencia.

Sin embargo, estos resultados nos han servido para realimentar nuestros proyectos futuros y ver que:

- El aspecto lúdico que aporte el robot (sentido del humor) es un factor motivador muy importante.
- La correcta parametrización del modelo de intervención pedagógica es importante de cara a no causar frustración en los alumnos durante el desarrollo de las actividades.
- Al igual que la parametrización del modelo de intervención pedagógica, resulta igualmente importante implementar un correcto escenario de aprendizaje que ayude al robot a identificar las dificultades en cada una de las actividades propuestas.

CAPÍTULO 11

CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se resumen los objetivos alcanzados en este proyecto, se presentan las conclusiones obtenidas y se proponen sugerencias para trabajos futuros.

12. Conclusiones

Como resultado de este trabajo fin de máster se han obtenido diferentes conclusiones expuestas a continuación:

- La implementación de ASIART llevada a cabo en este trabajo fin de máster sugiere que es posible realizar una intervención pedagógica afectiva adaptada a la idiosincrasia de la educación primaria y basada en una interacción oral con los alumnos a través de un robot tutor. En las distintas intervenciones, los alumnos han interactuado con NAO de forma natural y siguiendo las recomendaciones del robot. Esto respalda que es posible concebir un robot tutor afectivo que sirva de apoyo dentro de la enseñanza primaria.
- Se puede concluir además que ASIART admite la integración de Scratch como software educativo, y de NAO como robot tutor afectivo. Los resultados también indican que es posible realizar una intervención pedagógica afectiva diferente según el modelo de estudiante y según su estado emocional. Además ASIART ofrece un mecanismo capaz de construir una intervención pedagógica en un lenguaje de marcado genérico basándose en el modelo de estudiante y en un estado emocional.
- De igual forma los resultados sugieren la posibilidad de generar un modelo de clasificación capaz de identificar los estados emocionales de los alumnos de primaria, que condicionan de forma más relevante el carácter de la intervención educativa a través de las interacciones de teclado y de ratón. Dada la naturaleza no intrusiva de este método de identificación de estados emocionales, se ha observado que los alumnos mantienen su concentración en los ejercicios propuestos durante la experiencia.
- Además de la posibilidad de generar un fichero en un lenguaje de marcado que contiene la intervención pedagógica oportuna, la aproximación propuesta respalda la tesis de que es posible realizar un traductor de este lenguaje de marcado a acciones de un robot tutor, y concretamente, la posibilidad de traducir una intervención pedagógica de BML a acciones de NAO. Puesto que existen avatares virtuales en softwares educativos que hacen uso de lenguajes de marcado, existe la posibilidad de trasladar las acciones de dichos avatares virtuales a robots tutores mediante la traducción que se ha llevado a cabo en esta implementación de la arquitectura ASIART.
- Los resultados también sugieren que la metodología ASIART propuesta para el diseño de experiencias para elicitación de conocimiento sobre intervenciones pedagógicas ha sido adecuada a la hora de obtener la información necesaria para el diseño y la implementación de ASIART dentro de la educación primaria. Sin embargo, aún hay que seguir profundizando en la elaboración de dicha metodología.
- Después de la evaluación de ASIART, se apoya la tesis que la implementación llevada a cabo en este proyecto ha motivado a los alumnos a la hora de realizar los ejercicios, así como también les ha inspirado tranquilidad. Sin embargo, también se ha observado que un modelo pedagógico muy básico puede causar frustración, lo que hace que resulte vital diseñar un buen modelo, adecuado al escenario de aprendizaje y para el software educativo concreto.
- Otro elemento que puede causar frustración es una cantidad elevada de intervenciones en un corto espacio de tiempo, así como una incorrecta identificación del estado cognitivo-afectivo. Por ello ha sido necesario limitar las intervenciones cuando exista un cambio de estado cognitivo-afectivo, o cuando el alumno lleve un tiempo prudencial en el mismo estado cognitivo afectivo y que requiera de una intervención. Esto indica que resulta igual de importante diseñar un buen modelo emocional, que diseñar un buen modelo de intervención pedagógica.

- Por otro lado, no se ha avanzado en la definición de los estándares educativos.
- Los modelos de intervención pedagógica y de estudiante implementados en el prototipo están en una versión muy preliminar, y habría que seguir profundizando en ellos para conseguir un prototipo más estable.
- En el prototipo implementado en este proyecto, no se ha contado con un módulo capaz de identificar los escenarios de aprendizaje; en futuras versiones de prototipos sería conveniente diseñar e implementar tal módulo.
- Finalmente, este proyecto tiene múltiples dimensiones, en las que toca muchas disciplinas y campos de la Inteligencia Artificial y que requieren el conocimiento de muchas tecnologías, así como experiencias de usuario costosas en tiempo y recursos. De ahí que solo haya sido posible alcanzar los ambiciosos objetivos inicialmente planteados en cierta medida.

13. Trabajos Futuros

La investigación llevada a cabo en este trabajo fin de máster deja la puerta abierta a ciertos trabajos futuros, que cubren diferentes aspectos como son:

- Refinamiento del modelo de clasificación para la identificación de estados emocionales basado en las interacciones de teclado y de ratón: Para ello en primer lugar es necesario realizar una experiencia de recogida de datos de las interacciones de teclado y de ratón de mayor duración y con un mayor número de alumnos. Una vez obtenido el set de datos sería conveniente analizar de nuevo los métodos de minería de datos descritos, e incluir nuevos métodos para la generación del modelo de clasificación como puede ser el uso de la librería TensorFlow [TensorFlow] de Google.
- Implementación de métodos auxiliares no intrusivos de identificación de estados emocionales, tales como el análisis facial o el análisis de voz. Para ello se puede considerar el análisis de la librería TensorFlow de Google o CNTK [CNTK] (Computational Network Toolkit) de Microsoft.
- Dotar al robot tutor afectivo de la posibilidad de cotejar el estado emocional del alumno detectado desde el software educativo, mediante métodos tales como el análisis facial o el análisis de voz.
- Dotar al robot tutor afectivo de la capacidad de búsqueda e identificación de alumnos que requieren una intervención pedagógica, bien a través del reconocimiento facial, bien a través de cualquier otro método existente, con el fin de que pueda tutorizar simultáneamente a varios alumnos.
- Dotar a la plataforma de la posibilidad de identificar un mayor número de estados emocionales.
- Crear una herramienta para la generación de reglas, de cara a facilitar la implementación de un modelo de intervención pedagógica.
- Implementar un módulo para la identificación de escenarios de aprendizaje para el caso de Scratch.
- Implementar un generador de modelo de estudiante más completo que capture la idiosincrasia de los alumnos de primaria..
- Implementar un generador de modelo de intervención pedagógica más completo.
- Formalizar estándares educativos específicos para alumnos de primaria.

ANEXOS

Este capítulo recopila todos los anexos que complementan los capítulos anteriores.

Anexo 1: Formulario basado en BROMPT

En la siguiente figura se muestra el formulario utilizado por los observadores durante las sesiones de grabación para la toma de notas.

USUARIO:				
SOFTWARE:				
ESCUELA:				
CLASE:				
FECHA Y HORA DE INICIO:				
ID Estudiante	Hora	Comportamiento	Estado	Intervención

Anexo 2: WSDL del servicio web emocional

```

<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-
ws.dev.java.net. RI's version is JAX-WS RI 2.1.3-b02-. --><!-- Generated by JAX-WS RI at http://
jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3-b02-. --><definitions xmlns:soap="http://
schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://webservices.asiart/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://webservices.asiart/" name="ClassificationWSService">
<types>
<xsd:schema>
<xsd:import namespace="http://webservices.asiart/" schemaLocation="http://localhost:8080/
WSEmotionalAgent/ClassificationWS?xsd=1"></xsd:import>
</xsd:schema>
</types>
<message name="ClassifyEmotionalStatus">
<part name="arg0" type="xsd:string"></part>
<part name="arg1" type="tns:LogAnalyzedArray"></part>
</message>
<message name="ClassifyEmotionalStatusResponse">
<part name="return" type="xsd:string"></part>
</message>
<portType name="ClassificationWS">
<operation name="ClassifyEmotionalStatus" parameterOrder="arg0 arg1">
<input message="tns:ClassifyEmotionalStatus"></input>
<output message="tns:ClassifyEmotionalStatusResponse"></output>
</operation>
</portType>
<binding name="ClassificationWSPortBinding" type="tns:ClassificationWS">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"></soap:binding>
<operation name="ClassifyEmotionalStatus">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal" namespace="http://webservices.asiart/"></soap:body>
</input>
<output>
<soap:body use="literal" namespace="http://webservices.asiart/"></soap:body>
</output>
</operation>
</binding>
<service name="ClassificationWSService">
<port name="ClassificationWSPort" binding="tns:ClassificationWSPortBinding">
<soap:address location="http://localhost:8080/WSEmotionalAgent/ClassificationWS"></
soap:address>
</port>
</service>
</definitions>

```

Anexo 3: WSDL del servicio web de lenguaje de etiquetado

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-
ws.dev.java.net. RI's version is JAX-WS RI 2.1.3-b02-. --><!-- Generated by JAX-WS RI at http://
jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.3-b02-. --><definitions xmlns:soap="http://
schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://webservices.asiart/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://webservices.asiart/" name="MarkupWSService">
<types></types>
<message name="EmotionalMarkupConstructor">
<part name="arg0" type="xsd:string"/>
<part name="arg1" type="xsd:string"/>
</message>
<message name="EmotionalMarkupConstructorResponse">
<part name="return" type="xsd:string"/>
</message>
<portType name="MarkupWS">
<operation name="EmotionalMarkupConstructor" parameterOrder="arg0 arg1">
<input message="tns:EmotionalMarkupConstructor"/>
<output message="tns:EmotionalMarkupConstructorResponse"/>
</operation>
</portType>
<binding name="MarkupWSPortBinding" type="tns:MarkupWS">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
<operation name="EmotionalMarkupConstructor">
<soap:operation soapAction="">
<input>
<soap:body use="literal" namespace="http://webservices.asiart"/>
</input>
<output>
<soap:body use="literal" namespace="http://webservices.asiart"/>
</output>
</operation>
</binding>
<service name="MarkupWSService">
<port name="MarkupWSPort" binding="tns:MarkupWSPortBinding">
<soap:address location="http://localhost:8080/WSSMarkupAgent/MarkupWS"/>
</port>
</service>
</definitions>
```

BIBLIOGRAFÍA

Todas las referencias utilizadas en el desarrollo de este proyecto están recopiladas en este capítulo.

Bibliografía

- [Ahmed, 2005] Ahmed, I. (2015). *AIML Based Voice Enabled Artificial Intelligent Chatterbot*, (JANUARY). doi:10.14257/ijunesst.2015.8.2.36.
- [Alencar et al., 2014] Alencar, M., & Netto, J. F. (2014). *TUtor Collaborator Using Multi-Agent System*. *Collaboration Technologies and Social Computing*, 460(Communications in Computer and Information Science), 153–159. doi:10.1007/978-3-662-44651-5_14.
- [Alimisis, 2013] Alimisis, D. (2013). Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, 6(1), 63–71.
- [Almajano and Lopez-sanchez, 2013] Almajano, P., & Lopez-sanchez, M. (2013). *Conversational Structured Hybrid 3D Virtual Environments*.
- [Bailenson et al., 2007] Bailenson, J. N., Yee, N., Brave, S., Merget, D., & Koslow, D. (2007). *Virtual interpersonal touch: expressing and recognizing emotions through haptic devices*. *Human-Computer Interaction*, 22(November), 325–353. doi:10.1080/07370020701493509.
- [Beard et al., 2001] Beard, S., Gustavsson, C., Strindlund, L., Wiknertz, E., Huynh, Q., Marriott, A., & Stallo, J. (2001). *VHML*.
- [Beetz, 2013] Beetz, M. (2013). *Robot recommender system using affection-based episode ontology for personalization*. *2013 IEEE Ro-Man*, 155–160. doi:10.1109/ROMAN.2013.6628437.
- [Bertola and Patti, 2013] Bertola, F., & Patti, V. (2013). *Organizing Artworks in an Ontology-based Semantic Affective Space*.
- [Blazar, 2013] Blazar, D. (2013). *Harvard Graduate School of Education*.
- [Bisquerra, 2005] Bisquerra Alzina, R. (2005). La educación emocional en la formación del profesorado. *Revista Interuniversitaria de Formación Del Profesorado*, 19(3), 95–114.
- [Bousse et al., 2014] Bousse, E., Combemale, B., & Baudry, B. (2014). *Towards Scalable Multidimensional Execution Traces for xDSMLs*. *Proceedings of the Workshop on Model-Driven Engineering, Verification and Validation (MoDeVva '14)*.
- [Bradley and Lang, 1994] Bradley, M., & Lang, P. J. (1994). *Measuring Emotion: The Self-Assessment Semantic Differential Manikin and the*. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1), 49–59. doi:10.1016/0005-7916(94)90063-9.
- [Bredenkamp and Copple, 2009] Bredenkamp, S., & Copple, C. (2009). *Developmentally Appropriate Practice in Early Childhood Programs Serving Children from Birth through Age 8*. *Young Children*, (1994), 352. Retrieved from <http://www.naeyc.org/store/node/162>
- [Capponi et al., 2010] Capponi, M. F., Nussbaum, M., Marshall, G., & Lagos, M. E. (2010). *Pattern Discovery for the Design of Face-to-Face Computer-Supported Collaborative Learning Activities*. *Journal of Educational Technology & Society*, 13(2), p40–52.
- [Catlin and Robertson, 2012] Catlin, D., & Robertson, S. (2012). *Using Educational Robots to Enhance the Performance of Minority Students*, 12–21.

- [Choi et al., 2014] Choi, J., Cho, Y., Choi, J., & Choi, J. (2014). *A Layered Middleware Architecture for Automated Robot Services*, 2014.
- [Chuttur, 2009] Chuttur, M. (2009). *Overview of the Technology Acceptance Model: Origins , Developments and Future Directions*. *Sprouts: Working Papers on Information Systems*, 9(2009), 1–23. doi:10.1021/jf001443p
- [Claudia et al., 2007] Claudia, A., & Tade, R. (2007). *Robot Virtual en Orientación Vocacional*, (02), 1–18.
- [CNTK] <http://www.cntk.ai>
- [Cocea and Weibelzahl, 2009] Cocea, M., & Weibelzahl, S. (2009). *Log file analysis for disengagement detection in e-Learning environments*. *User Modelling and User-Adapted Interaction* (Vol. 19). doi:10.1007/s11257-009-9065-5.
- [Coradeschi and Saffiotti, 2006] Coradeschi, S., & Saffiotti, A. (2006). *Symbiotic robotic systems: humans, robots, and smart environments*. *IEEE Intelligent Systems*, 21(3), 82–84. doi: 10.1109/MIS.2006.59.
- [Corrigan et al., 2013] Corrigan, L. J., Peters, C., & Castellano, G. (2013). *Identifying task engagement: Towards personalised interactions with educational robots*. *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 655–658. doi: 10.1109/ACII.2013.114.
- [Cruz-Lara et al., 2014] Cruz-Lara, S., Denis, A., & Bellalem, N. (2014). *Linguistic and multilingual issues in virtual worlds and serious games: a general review*. *Journal For Virtual Worlds ...*, 7(1), 1–23. Retrieved from <http://journals.tdl.org/jvwr/index.php/jvwr/article/view/7084>
- [Duit and Tragust, 2013] Duit, R., & Treagust, D. F. (2013). *A powerful framework for improving science teaching and learning*. *Journal of Chemical Information and Modeling*, 53(6), 1689–1699.
- [Fong et al., 2002] Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2002). *A survey of socially interactive robots*. *Robotics and Autonomous Systems*, 42(3-4), 143–166. doi:10.1016/S0921-8890(02)00372-X.
- [Fridin and Belokopytov, 2013] Fridin, M., & Belokopytov, M. (2013). *Computers in Human Behavior Acceptance of socially assistive humanoid robot by preschool and elementary school teachers*. *Computers in Human Behavior*, 33, 23–31. doi:10.1016/j.chb.2013.12.016.
- [Gascueña et al., 2004] Gascueña, J. M., Fernández-caballero, A., & González, P. (2004). *Ontologías del modelo del alumno y del modelo del dominio en sistemas de aprendizaje adaptativos y colaborativos*.
- [Gelin et al., 2010] Gelin, R., D’Alessandro, C., & Le, Q. (2010). *Towards a storytelling humanoid robot*. *AAAI Fall Symposium ...*, 137–138. Retrieved from <http://www.aaai.org/ocs/index.php/FSS/FSS10/paper/viewFile/2169/2747>.
- [Goc et al., 2007] Goc, B., Kufel, M., & Puchacz, M. (2007). *Rule languages used in Tutorial chatbots programming*, 7, 163–170.
- [Grassi, 2009] Grassi, M. (2009). *Developing HEO Human Emotions Ontology*. *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5707 LNCS, 244–251. doi:10.1007/978-3-642-04391-8_32.

- [Gustavsson et al., 2002] Gustavsson, C., Strindlund, L., & Wiknertz, E. (2002). *Verification, Validation and Evaluation of the Virtual Human Markup Language (VHML)*.
- [Hammer et al., 2015] Hammer, S., Rist, T., Kastrinaki, S., Hondrou, C., Raouzaiou, A., Karpouzis, K., & Kollias, S. (2015). *Design of a Lifestyle Recommender system for the Elderly : Requirement Gatherings in Germany and Greece*.
- [Hastings et al., 2011] Hastings, J., Ceusters, W., Smith, B., & Mulligan, K. (2011). The emotion ontology: Enabling interdisciplinary research in the affective sciences. *Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6967 LNAI, 119–123. doi:10.1007/978-3-642-24279-3_14.
- [Hastings et al., 2012] Hastings, J., Ceusters, W., Mulligan, K., & Smith, B. (2012). Annotating affective neuroscience data with the Emotion Ontology, 1–5.
- [Heckmann et al., 2007] Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., & KrAöner, A. (2007). The user model and context ontology GUMO revisited for future Web 2.0 extensions. *CEUR Workshop Proceedings*, 298, 42–51.
- [Hernández Orallo et al., 2004] Hernández Orallo, J., Ramírez Quintana M. J., Ferri Ramírez C. (2004). *Introducción a la minería de datos*, pp. 12, ISBN: 84-205-4091-9.
- [Hernando, 2015] Hernando, A. (2015). *Viaje a la escuela del siglo XXI*.
- [Heylen et al., 2008] Heylen, D., Kopp, S., Marsella, S., & Vilhjalmsson, H. (2008). *The Next Step Towards a Functional Markup Language. Artificial Intelligence*.
- [Irani and Fayyad, 1993] Irani, K. B. and Fayyad, U. M. (1993). *Multi- interval discretization of continuous-valued attributes for classification learning*. *Proc10th IntConfMachine Learning*, pages 194–201.
- [Ismail et al., 2012] Ismail, L. I., Shamsudin, S., Yussof, H., Hanapiah, F. A., & Zahari, N. I. (2012). *Estimation of Concentration by Eye Contact Measurement in Robot-based Intervention Program with Autistic Children*. *Procedia Engineering*, 41(Iris), 1548–1552. doi:10.1016/j.proeng.2012.07.348.
- [Ito, 2005] Ito, K. (2005). *Introducing Multimodal Character Agents Into Existing Web Applications*. *Www*, 966–967. doi: 10.1145/1062745.1062821.
- [Jimenez et al., 2015] Jimenez, F., Yoshikawa, T., Furuhashi, T., & Kanoh, M. (2015). *AN EMOTIONAL EXPRESSION MODEL FOR EDUCATIONAL-SUPPORT ROBOTS*, 5(1), 51–57.
- [Jovanova et al., 2009] Jovanova, B., Preda, M., & Preteux, F. (2009). *The Role of Interoperability in Virtual Worlds, Analysis of the Specific Cases of Avatars*. *Journal of Virtual Worlds Research*, 2(3). Retrieved from <http://journals.tdl.org/jvwr/article/view/672>.
- [Kalelioğlu and Gülbahar, 2014] Kalelioğlu, F., & Gülbahar, Y. (2014). *The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective*. *Informatics in Education*, 13(1), 33–50.
- [Kamei et al., 2011] Kamei, K., Ikeda, T., Kidokoro, H., Shiomi, M., Utsumi, A., Shinozawa, K., ... Hagita, N. (2011). *Effectiveness of cooperative customer navigation from robots around a retail shop*. *Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/SocialCom 2011, (August 2015)*, 235–241. doi:10.1109/PASSAT/SocialCom.2011.173.

- [Kaur and Chaudhary, 2015] Kaur, G., & Chaudhary, D. (2015). Semantic Web : A Boon for E-learning, 4(7), 484–486. doi:10.17148/IJARCCE.2015.47110.
- [Keren et al., 2012] Keren, G., Ben-David, A., & Fridin, M. (2012). *Kindergarten assistive robotics (KAR) as a tool for spatial cognition development in pre-school education*. IEEE International Conference on Intelligent Robots and Systems, 1084–1089. doi:10.1109/IROS.2012.6385645.
- [Khan et al., 2013] Khan, I. A., Brinkman, W. P., & Hierons, R. (2013). *Towards estimating computer users' mood from interaction behaviour with keyboard and mouse*. Frontiers of Computer Science, 7(6), 943–954. doi:10.1007/s11704-013-2331-z.
- [Khanna and Sasikumar, 2010] Khanna, P., & Sasikumar, M. (2010). *Recognising Emotions from Keyboard Stroke Pattern*. International Journal of Computer Applications, 11(9), 1–5. doi:10.5120/1614-2170.
- [Kononenko, 1994] Kononenko, I. (1994). *Estimating attributes: Analysis and extensions of relief*.
- [Koo et al., 2011] Koo, S., Park, K., Kim, H., & Kwon, D. (2011). *A Dual-Layer User Model based Cognitive System for User-Adaptive Service Robots*. Symposium A Quarterly Journal In Modern Foreign Literatures, 59–64.
- [Kopp et al., 2006] Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., ... Vilhjálmsson, H. (2006). *Towards a common framework for multimodal generation: The behavior markup language*. Intelligent Virtual Agents, 4133, 205–217. doi:10.1007/11821830.
- [Liu et al., 2010] Liu, I. F., Chen, M. C., Sun, Y. S., Wible, D., & Kuo, C. H. (2010). *Extending the TAM model to explore the factors that affect Intention to Use an Online Learning Community*. Computers and Education, 54(2), 600–610. doi:10.1016/j.compedu.2009.09.009
- [Liu et al., 2015] Liu, W., & Dong, W. (2015). *Research on improved dialogue model, (Etmhs)*, 827–831.
- [Lohse et al., 2012] Lohse, M., & Welbergen, H. Van. (2012). *Designing Appropriate Feedback for Virtual Agents and Robots. ... Robot Feedback in Human-Robot ...*. Retrieved from <http://pub.uni-bielefeld.de/download/2537561/2537564>.
- [López García, 2012] López García, J. C. (2012). *Identificación y regulación de emociones con Scratch. Tendencias emergentes en Educación con TIC*.
- [López García, 2014] López García, J. C. (2014). *ACTIVIDADES DE AULA CON SCRATCH QUE FAVORECEN EL USO DEL PENSAMIENTO ALGORÍTMICO*.
- [Makatchev Maxim et al., 2000] Makatchev Maxim, T. S. K. (2000). *Human-Robot Interface Using Agents Communicating in an XML-Based Markup Language*.
- [Marriot et al., 2006] Marriott, A., Holic, A., & Reid, D. (2006). *The Multi-modal Mentor System. IUI 2006 Workshop on Effective Multimodal Dialogue Interfaces*, 49.
- [Marriott, 2011] Marriott, A. (2001). *VHML–Virtual Human Markup Language. Online*, [Http://www.Interface.Computing.Edu.au/document/VHML,\(Xml\),2002](Http://www.Interface.Computing.Edu.au/document/VHML,(Xml),2002).

- [Marriott and Shortland-Jones, 2003] Marriott, A., & Shortland-Jones, B. (2003). *The mentor system*. Workshop Proceedings of 11th International Conference on Artificial Intelligence in Education, 18.
- [Meftah, 2013] Meftah, I. T. (2013). *Docteur en Sciences Tayari Meftah*.
- [Mori et al., 2003] Mori, K., Jatowt, A., & Ishizuka, M. (2003). *Enhancing conversational flexibility in multimodal interactions with embodied life like agent*. Proceedings of the International Conference on Intelligent Multimedia and Distant Education, 270–272. doi: 10.1145/604045.604096.
- [NAO] Aldebaran. NAO robot: intelligent and friendly companion. <https://www.aldebaran.com/en/humanoid-robot/nao-robot>.
- [Nourbakhsh et al., 1999] Nourbakhsh, I. R., Bobenage, J., Grange, S., Lutz, R., Meyer, R., & Soto, A. (1999). *An affective mobile robot educator with a full-time job*. Artificial Intelligence, 114(1-2), 95–124. doi:10.1016/S0004-3702(99)00027-2.
- [Nozawa et al., 2004] Nozawa, Y., Dohi, H., Iba, H., & Ishizuka, M. (2004). Humanoid robot presentation controlled by multimodal presentation markup language MPML. RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759), 153–158. doi:10.1109/ROMAN.2004.1374747.
- [Núñez et al., 2014] Núñez, P. V., & Quingatuña, A. (2014). *Robot socialmente asistencial y de interacción multimodal autónoma*, 5(9), 15–24.
- [Ocumpaugh, 2012] Ocumpaugh, J. (2012). *Baker-Rodrigo Observation Method Protocol (BROMP) 1.0*.
- [Ocumpaugh and College, 2014] Ocumpaugh, J., & College, T. (2014). *Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) 2.0 Technical and Training Manual*.
- [Paiva et al., 2012] Paiva, A., Leite, I., & Ribeiro, T. (2012). *Emotion Modelling for Social Robots*. Ict.Usc.Edu. Retrieved from <http://ict.usc.edu/~gratch/CSCI534/ACII-Handbook-Robots.pdf>
- [Pandzic and Forchheimer, 2003] Pandzic, I. S., & Forchheimer, R. (2003). *MPEG-4 Facial Animation*. The Standard, Implementation and Applications (John Wiley & Sons, LTD, 2002) (Vol. 13). doi: 10.1002/ima.10072.
- [Paper et al., 2014] Paper, C., Prestes, E., Federal, U., Grande, R., Rama, S., Pontif, F., ... Grande, R. (2014). *A Method for designing Dialogue Systems by using Ontologies Ewerton, (SEPTEMBER)*.
- [Pellegrini and Blatchford, 2000] Pellegrini, A. D., & Blatchford, P. (2000). *The child at school: Interactions with peers and teachers*. *The Child at School: Interactions with Peers and Teachers*. doi:10.4324/9781315726953
- [Picard et al., 2001] Picard, R. W., Vyzas, E., & Healey, J. (2001). *Toward machine emotional intelligence: analysis of affective\physiological state*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10), 1175–1191. doi:10.1109/34.954607.
- [Porayska-Pomsta et al., 2013] Porayska-Pomsta, K., Mavrikis, M., D’Mello, S., Conati, C., & Baker, R. S. J. D. (2013). *Knowledge elicitation methods for affect modelling in education*. *International Journal of Artificial Intelligence in Education*, 22(3), 107–140. doi:10.3233/JAI-130032.

- [Prendinger et al., 2004] Prendinger, H., Descamps, S., & Ishizuka, M. (2004). *MPML: A markup language for controlling the behavior of life-like characters*. Journal of Visual Languages and Computing, 15(2), 183–203. doi:10.1016/j.jvlc.2004.01.001
- [Prendinger et al., 2011] Prendinger, H., Ullrich, S., Nakasone, A., & Ishizuka, M. (2011). *MPML3D: Scripting agents for the 3D internet*. IEEE Transactions on Visualization and Computer Graphics, 17(5), 655–668. doi:10.1109/TVCG.2010.66
- [Rajput and Vijayavargiya, 2015] Rajput, S., & Vijayavargiya, P. (2015). *Objective of Keystroke Dynamics for Identifying Emotional State*, 6(1), 632–636.
- [Regueiro, 2009] Regueiro, C. V. (2009). *Teaching Digital Systems in the Context of the New European Higher Education Area : A Practical Experience*. Architecture, 52(4), 513–523.
- [Resch et al., 2015] Resch, B., Sudmanns, M., Sagl, G., Summa, A., Zeile, P., & Exner, J.-P. (2015). Crowdsourcing Physiological Conditions and Subjective Emotions by Coupling Technical and Human Mobile Sensors. GI_Forum 2015 - Geospatial Minds for Society, pending.
- [Ribeiro et al., 2013] Ribeiro, T., Aylett, R., & Paiva, A. (2013). *I ' m the Mayor : A Robot Tutor in Encities-2 (Demonstration)*, 2, 1675–1676.
- [Riccardi, 2014] Riccardi, G. (2014). *Towards Healthcare Personal Agents*.
- [Riggs et al., 2006] Riggs, N. R., Greenberg, M. T., Kusché, C. A., & Pentz, M. A. (2006). The Mediation Role of Neurocognition in the Behavioral Outcomes of a Social-Emotional Prevention Program in Elementary School Students: Effects of the PATHS Curriculum. Prevention Science, 7(1), 91–102. doi:10.1007/s11121-005-0022-1.
- [Robins et al., 2004] Robins, B., Dautenhahn, K., & Dubowski, J. (2004). *Investigating autistic children's attitudes towards strangers with the theatrical robot - a new experimental paradigm in human-robot interaction studies*. RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No. 04TH8759), 557–562. doi:10.1109/ROMAN.2004.1374821.
- [Roesch et al., 2011] Roesch, E. B., Korsten, N., Fragopanagos, N. F., Taylor, J. G., Grandjean, D., & Sander, D. (2011). *Biological computational constraints to the psychological modelling of emotion*. doi:10.1007/978-3-642-15184-2.
- [Rosenthal et al., 2009] Rosenthal, S., Dey, A. K., & Veloso, M. (2009). *How robots' questions affect the accuracy of the human responses*. Proceedings - IEEE International Workshop on Robot and Human Interactive Communication, 1137–1142. doi:10.1109/ROMAN.2009.5326291.
- [Russell et al., 2014] Russell, E., Stroud, A., Christian, J., Ramgoolam, D., & Williams, A. B. (2014). *SMILE : A Portable Humanoid Robot Emotion Interface*.
- [Saerbeck et al., 2010] Saerbeck, M., Schut, T., Bartneck, C., & Janse, M. D. (2010). *Expressive Robots in Education Varying the Degree of Social Supportive Behavior of a Robotic Tutor*, 1613–1622.
- [Saeyor et al., 2003] Saeyor, S., Uchiyama, K., & Ishizuka, M. (2003). *Multimodal Presentation Markup Language on Mobile Phones*.

- [Salam and Chetouani, 2015] Salam, H., & Chetouani, M. (2015). *A Multi-Level Context-Based Modeling of Engagement in Human-Robot Interaction A Multi-Level Context-Based Modeling of Engagement*.
- [Salmeron-majadas, 2014] Salmeron-majadas, S. (2014). *Affective standards-based modeling in educational contexts from mining multimodal data sources State of the art*.
- [Salmeron-majadas et al., 2014] Salmeron-majadas, S., Santos, O. C., & Boticario, J. G. (2014). *Exploring indicators from keyboard and mouse interactions to predict the user affective state*, 365–366.
- [Scratch] Massachusetts Institute of Technology. Scratch - Imagine, Program, Share. <https://scratch.mit.edu>.
- [Schröder et al., 2011] Schröder, M., Baggia, P., Burkhardt, F., Pelachaud, C., Peter, C., & Zovato, E. (2011). *EmotionML – an upcoming standard for representing emotions and related states*.
- [Schuller et al., 2014] Schuller, B., Baron-cohen, S., Rielly, H. O., Robinson, P., Davies, I., Golan, O., ... Camurri, A. (2014). *ASC-Inclusion : Integrated Internet-Based Environment for Social Inclusion of Children with Autism Spectrum Conditions*.
- [Selim, 2003] Selim, H. M. (2003). *An empirical investigation of student acceptance of course websites. Computers and Education*, 40(4), 343–360. doi:10.1016/S0360-1315(02)00142-2
- [Shelton, et al., 2012] Shelton, C. D. (2012). *ROBOT-OBJECT INTERACTION LANGUAGE*.
- [Sieg et al., 2007] Sieg, A., Mobasher, B., & Burke, R. (2007). *Learning Ontology-Based User Profiles : A Semantic Approach to Personalized Web Search. Computational Intelligence*, 8, 1–12.
- [Tanaka and Matsuzoe, 2012] Tanaka, F., & Matsuzoe, S. (2012). *Children Teach a Care-Receiving Robot to Promote Their Learning: Field Experiments in a Classroom for Vocabulary Learning. Journal of Human-Robot Interaction*, 1(1), 78–95. doi:10.5898/JHRI.1.1.Tanaka
- [Tang et al., 2015] Tang, D., Yusuf, B., Botzheim, J., Kubota, N., & Chan, C. S. (2015). *A novel multimodal communication framework using robot partner for aging population. Expert Systems with Applications*, 42(9), 4540–4555. doi:10.1016/j.eswa.2015.01.016.
- [TensorFlow] <https://www.tensorflow.org>.
- [Tsutsui et al., 2000] Tsutsui, T., Saeyor, S., & Ishizuka, M. (2000). *MPML : A Multimodal Presentation Markup Language with Character Agent Control Functions Features of Multimodal Presentation Markup Language MPML*, 1(Takebayashi 94).
- [Umrani-khan, 2000] Umrani-khan, F. (2000). *Computer Attitude and Fluency : A Study of Elementary School Students Framework for building computer fluency amongst school students*, (1990).
- [Valle and Salgado, 2013] Valle, J. E. M., & Salgado, V. C. (2013). *Pensamiento Lógico Matemático Con Scratch En Nivel Básico. Vínculos*, 9(1), 87–95.
- [Veena Vijayan, 2014] Veena Vijayan, V. (2014). *Investigation about the Impact of Robots in Educational and Medical Field*, 3(10), 436–441.

- [Vouloutsi et al., 2014] Vouloutsi, V., Munoz, M. B., Grechuta, K., Lallee, S., Duff, A., Llobet, J. P., & Verschure, P. F. M. J. (2014). *A new biomimetic approach towards educational robotics : the Distributed Adaptive Control of a Synthetic Tutor Assistant*.
- [Wilson and Moffat, 2010] Wilson, A., & Moffat, D. C. (2010). *Evaluating Scratch to introduce younger schoolchildren to programming. Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group, 64–75*.
- [Yang et al., 2004] Yang, Z., Ishizuka, M., & Engineering, C. (2004). *MPML-FLASH: A Multimodal Presentation Markup Language with Character Agent Control in Flash Medium*. Zhenglu Yang Mitsuru Ishizuka Department of Information and Communication Engineering Graduate School of Information Science and Technology University of T.
- [Zimmermann et al., 2003] Zimmermann, P., Guttormsen, S., Danuser, B., & Gomez, P. (2003). *Affective Computing – Measuring Mood with Mouse and Keyboard, 2–5*.
- [Zong et al., 2000] Zong, Y., Dohi, H., Prendinger, H., & Ishizuka, M. (2000). *Emotion Expression Function in Multimodal Presentation*. *Artificial Intelligence*, 57–64.