



**Facultad de Ciencias de la Administración**

**Carrera de Ingeniería en Ciencias de la  
Computación**

**PROTOTIPO DE MIDDLEWARE BASADO EN  
MODELS@RUN.TIME DE AUTOCONSCIENCIA  
DISTRIBUIDA Y COLECTIVA PARA  
SISTEMAS IOT**

**Trabajo de titulación previo a la obtención del  
grado de Ingeniero en Ciencias de la  
Computación**

**Autor:**

Alex Fernando Coro Chimbo.

**Director:**

Ing. Lenin Xavier Erazo Garzón MSc.

**Cuenca – Ecuador**

**2024**

## **DEDICATORIA**

A mis padres y hermano, quienes fueron los pilares fundamentales en mi vida académica, agradezco infinitamente su amor y apoyo. Este trabajo de titulación es un tributo a su sacrificio y constante aliento.

## **AGRADECIMIENTO**

Mi sincero agradecimiento al ingeniero Lenin Erazo, director del presente trabajo, por su orientación, paciencia y dedicación. Asimismo, a la ingeniera María Inés Acosta por su excepcional asesoramiento y contribución. También agradezco a la Universidad del Azuay por ofrecer el entorno propicio para mi formación. A mi familia y amigos agradezco su compañía durante mi vida académica.

## Índice de Contenidos

<b>DEDICATORIA</b> .....	i
<b>AGRADECIMIENTO</b> .....	ii
<b>Índice de Contenidos</b> .....	iii
<b>Índice de Figuras</b> .....	v
<b>Índice de Tablas</b> .....	vi
<b>RESUMEN</b> .....	vii
<b>ABSTRACT</b> .....	vii
<b>1. Introducción</b> .....	1
<b>1.1 Objetivos</b> .....	1
<b>1.1.1 Objetivo general</b> .....	1
<b>1.1.2 Objetivos específicos</b> .....	1
<b>2. Revisión de literatura</b> .....	1
<b>2.1 Marco teórico</b> .....	1
<b>2.1.1 Internet de las cosas</b> .....	2
<b>2.1.1.1 Dominios</b> .....	2
<b>2.1.1.2 Arquitectura</b> .....	2
<b>2.1.2 Autoconsciencia y sistemas autoconscientes</b> .....	3
<b>2.1.2.1 Aspectos clave de la autoconsciencia</b> .....	4
<b>2.1.2.2 Autoconsciencia colectiva</b> .....	4
<b>2.1.2.3 Aspectos de autoconsciencia</b> .....	4
<b>2.1.3 Ingeniería dirigida por modelos</b> .....	5
<b>2.1.4 Desarrollo basado en modelos</b> .....	5
<b>2.1.5 Modelos en tiempo de ejecución</b> .....	5
<b>2.2 Estado del arte</b> .....	6
<b>3. Materiales y Métodos</b> .....	7
<b>3.1 Herramientas</b> .....	7
<b>3.1.1 JavaScript</b> .....	7
<b>3.1.1 Node.js</b> .....	7
<b>3.1.2 PostgreSQL</b> .....	8
<b>3.1.3 Docker</b> .....	8
<b>3.2 Metodología</b> .....	8

<b>4. Middleware basado en models@run.time de autoconsciencia distribuida y colectiva .....</b>	<b>9</b>
<b>4.1 Visión general del middleware.....</b>	<b>9</b>
<b>4.2 Arquitectura del middleware.....</b>	<b>9</b>
<b>4.3 Implementación de los componentes del middleware .....</b>	<b>11</b>
<b>4.3.1 Componentes de la aplicación supervisora .....</b>	<b>11</b>
<b>4.3.2 Componentes de la aplicación implementadora.....</b>	<b>13</b>
<b>4.3.3 Visor de autoconsciencia.....</b>	<b>18</b>
<b>5. Estudio de caso.....</b>	<b>20</b>
<b>5.1 Especificación de requerimientos de autoconsciencia del escenario de IoT ..</b>	<b>20</b>
<b>5.2 Implementación del estudio de caso .....</b>	<b>23</b>
<b>5.2.1 Implementación de la estructura de Docker.....</b>	<b>23</b>
<b>5.2.2 Implementación de los artefactos de autoconsciencia distribuida y colectiva.....</b>	<b>25</b>
<b>5.3 Operación y resultados de la infraestructura de autoconsciencia distribuida y colectiva.....</b>	<b>26</b>
<b>6. Conclusiones y Trabajo Futuro.....</b>	<b>32</b>

## Índice de Figuras

<b>Figura 1</b> Metodología de la investigación de Gorschek et al. (2006) aplicada al Trabajo de Titulación .....	9
<b>Figura 2</b> Arquitectura del middleware de autoconsciencia distribuida y colectiva.....	10
<b>Figura 3</b> Fragmento StartApp.js.....	11
<b>Figura 4</b> Fragmento Observador.js.....	12
<b>Figura 5</b> Fragmento PlanificadorTareas.js .....	12
<b>Figura 6</b> Fragmento Orquestador.js.....	13
<b>Figura 7</b> Fragmento StartApp.js.....	13
<b>Figura 8</b> Fragmento Recolector.js .....	14
<b>Figura 9</b> Fragmento EjecutorTareasAutoconsciencia.js.....	14
<b>Figura 10</b> Fragmento ImplementadorFunciones.js.....	15
<b>Figura 11</b> Fragmento ImplementadorServicios.js .....	15
<b>Figura 12</b> Fragmento ImplementadorBD.js .....	16
<b>Figura 13</b> Fragmento ImplementadorDemoniosPreReflexivos.js .....	17
<b>Figura 14</b> Fragmento ModeloLSP.js .....	17
<b>Figura 15</b> Fragmento ImplementadorDemoniosReflexivos.js .....	18
<b>Figura 16</b> Fragmento HTML visor de autoconsciencia. ....	19
<b>Figura 17</b> Fragmento CSS visor de autoconsciencia.....	19
<b>Figura 18</b> Fragmento JavaScript visor de autoconsciencia. ....	19
<b>Figura 19</b> Contenedores Docker para cada nodo del sistema IoT.....	23
<b>Figura 20</b> Configuración de archivo Dockerfile de la aplicación implementadora.....	24
<b>Figura 21</b> Configuración de archivo Dockerfile de la aplicación supervisora. ....	24
<b>Figura 22</b> Configuración de servicios y su comunicación en Docker-compose.yml. ....	25
<b>Figura 23</b> Implementación de artefactos de autoconsciencia distribuida y colectiva de la aplicación supervisora. ....	25
<b>Figura 24</b> Implementación de artefactos de autoconsciencia distribuida y colectiva del nodo IoT Gateway.....	26
<b>Figura 25</b> Implementación de artefactos de autoconsciencia distribuida y colectiva del nodo Fog. ....	26
<b>Figura 26</b> Datos monitoreados sobre la temperatura ambiente actual del dormitorio.....	27
<b>Figura 27</b> Datos almacenados sobre la temperatura ambiente actual del dormitorio.....	28
<b>Figura 28</b> Resultados de la ejecución del demonio responsable del proceso pre-reflexivo PID1. ....	28
<b>Figura 29</b> Almacenamiento de métricas e indicadores en la base de datos por el demonio responsable del proceso pre-reflexivo PID1. ....	28
<b>Figura 30</b> Datos monitoreados sobre la concentración actual de CO en la cocina.....	29
<b>Figura 31</b> Datos almacenados sobre la concentración actual de CO en la cocina.....	29
<b>Figura 32</b> Resultados de la ejecución del demonio responsable del proceso pre-reflexivo PID2. ....	30
<b>Figura 33</b> Almacenamiento de métricas e indicadores en la base de datos por el demonio responsable del proceso pre-reflexivo PID2. ....	30
<b>Figura 34</b> Resultados de la ejecución del demonio responsable del proceso reflexivo-colectivo PID3. ....	31
<b>Figura 35</b> Almacenamiento de indicadores e información asociada en la base de datos por el demonio responsable del proceso reflexivo-colectivo PID3.....	31
<b>Figura 36</b> Visor de autoconsciencia en operación.....	32

## Índice de Tablas

<b>Tabla 1</b> Descripción de los procesos pre-reflexivos y reflexivos a implementar en el estudio de caso.....	21
<b>Tabla 2</b> Descripción de métricas a utilizar por los métodos de los procesos de autoconsciencia.....	22
<b>Tabla 3</b> Descripción de los criterios de decisión y sus umbrales para evaluar cualitativamente los aspectos de autoconsciencia. ....	22
<b>Tabla 4</b> Descripción de recursos de implementación para procesos de autoconsciencia. ....	22

## RESUMEN

El desarrollo de software para sistemas IoT presenta importantes desafíos en términos de dinamismo e incertidumbre operacional, debido a que los enfoques tradicionales se centran en incorporar los requerimientos de software en tiempo de diseño, siendo complejo gestionar la evolución de estos sistemas. Un enfoque adecuado es incorporar la autoconsciencia computacional conjuntamente con los modelos en tiempos de ejecución, más aún cuando esta alternativa ha sido poco abordada, especialmente en lo relacionado a la autoconsciencia colectiva. Este trabajo de titulación propone construir un prototipo de middleware basado en modelos en tiempo de ejecución para la implementación distribuida de artefactos de software de autoconsciencia colectiva para sistemas IoT, utilizando el entorno de ejecución Node.js. Los artefactos de software implementados encapsulan una lógica que ejecuta procesos de autoconsciencia pre-reflexivos y reflexivos para evaluar mediante métricas el estado del sistema y su entorno; y, en caso de ser necesario, determinar recomendaciones correctivas.

**Palabras clave:** Autoconsciencia colectiva, Ingeniería Dirigida por Modelos (MDE), Internet de las Cosas (IoT), Middleware, Modelos en tiempo de ejecución (models@run.time).

## ABSTRACT

The development of software for IoT systems presents significant challenges in terms of dynamism and operational uncertainty, considering that traditional approaches focus on incorporating software requirements at design time, being complex to manage the evolution of these systems. A suitable approach is to integrate computational self-awareness into IoT systems through runtime models, even more so when this alternative has been little addressed, especially regarding collective self-awareness. This degree work proposes to build a prototype middleware based on runtime models for the distributed implementation of collective self-awareness software artifacts for IoT systems using the Node.js platform. The implemented software artifacts encapsulate programming logic that executes pre-reflexive and reflexive self-awareness processes to evaluate the state of the system and its environment by metrics and, if necessary, determine corrective recommendations.

**Keywords:** Collective self-awareness, Internet of Things (IoT), Middleware, Model-Driven Engineering (MDE), Runtime models (models@run.time).