



Facultad de Ciencias de la Administración

**Carrera de Ingeniería de Sistemas y
Telemática**

**PROTOTIPO DE MIDDLEWARE BASADO EN
MODELS@RUN.TIME PARA
IMPLEMENTACIÓN AUTOMÁTICA DE
ARTEFACTOS DE SOFTWARE EN SISTEMAS
DISTRIBUIDOS. ESTUDIO DE CASO:
MONITOREO DE SISTEMAS IOT**

**Trabajo de titulación previo a la obtención del
grado de Ingeniero de Sistemas y Telemática**

Autor:

Brandon Iván Llivisaca Largo

Director:

Ing. Lenín Xavier Erazo Garzón

Cuenca – Ecuador

2024

DEDICATORIA

Este trabajo de titulación va dedicado a mí madre, Carmen Largo, que, a pesar de cada tropiezo que se presentó en este trayecto, nunca me dejó solo, siempre me forjo con su amor incondicional, su sabiduría y su dedicación, inspirándome a ser la persona que soy hoy en día.

AGRADECIMIENTO

Agradezco de manera especial a Dios y a todas las personas que me apoyaron constantemente a lograr esta meta en mi vida. Sus palabras de aliento fueron un faro de luz en los momentos desafiantes. De igual manera, expreso mi sincero agradecimiento a mi director de tesis, Ing. Lenin Erazo, por su orientación y apoyo constante a lo largo de este proceso. Su sabiduría y orientación han sido fundamentales para el éxito de esta tesis.

Índice de Contenidos

Dedicatoria.....	i
Agradecimiento	ii
Índice de Contenidos.....	iii
Índice de Figuras	iv
Índice de Tablas.....	v
Resumen.....	vi
Abstract.....	vi
1. Introducción.....	1
1.1. Objetivos	1
1.1.1. Objetivo general	1
1.1.2. Objetivos específicos.....	1
2. Marco Teórico	1
2.1. Sistemas Distribuidos.....	1
2.2. Internet de las Cosas (IoT)	2
2.3. Ingeniería Dirigida por Modelos (MDE).....	2
2.4. Desarrollo dirigido por modelos (MDD)	3
2.5. Modelos en tiempo de ejecución (models@run.time).....	3
3. Estado del Arte	3
4. Métodos y Herramientas	4
4.1. Metodología	4
4.2. Herramientas	5
4.2.1. JavaScript	5
4.2.2. Node.js	6
4.2.3. MQTT	6
4.2.4. PostgreSQL	6
4.2.5. Docker.....	7
5. Middleware basado en models@run.time para la implementación automática de artefactos de software de monitoreo para sistemas IoT	7
5.2. Componentes de la Arquitectura del middleware.....	9
5.2.1. Componentes de la aplicación supervisora	9
5.2.2. Componentes de la aplicación implementadora.....	12
6. Estudio de caso	18
6.1. Especificación de requerimientos de monitoreo del escenario de IoT.....	18
6.2. Implementación del estudio de caso.....	19
6.2.1. Implementación de la estructura de Docker	19
6.2.2. Implementación de los artefactos de monitoreo del sistema IoT	22
6.3. Operación y resultados del prototipo de middleware	24
7. Conclusiones	27
8. Referencias.....	27

Índice de Figuras

Figura 1. Metodología de la investigación de Gorschek et al. (2006) aplicada al Trabajo de Titulación.....	5
Figura 2. Arquitectura del middleware para la implementación automática de los artefactos de software de monitoreo de un sistema IoT	8
Figura 3. Artefacto StartApp.js	10
Figura 4. Artefacto Observador.js	10
Figura 5. Artefacto PlanificadorTareas.js.....	11
Figura 6. Artefacto Orquestador.js	11
Figura 7. Artefacto StartApp.js	13
Figura 8. Artefacto Recolector.js.....	13
Figura 9. Artefacto SuscriptorTopicos.js.....	14
Figura 10. Artefacto DistribuidorMensajesTopicos.js	14
Figura 11. Artefacto ImplementadorServicios.js	15
Figura 12. Artefacto EjecutorTareasMonitoreo.js.....	15
Figura 13. Artefacto ImplementadorApis.js	16
Figura 14. Artefacto ImplementadorAplicaciones.js	16
Figura 15. Artefacto ImplementadorBD.js.....	17
Figura 16. Artefacto GestorManipulacionBD.js	17
Figura 17 Escenario ilustrativo de IoT.	19
Figura 18 Configuración del archivo Dockerfile de la aplicación supervisora.	20
Figura 19 Configuración del archivo Dockerfile de la aplicación implementadora.	20
Figura 20. Configuración del archivo Docker Compose.....	21
Figura 21 Contenedores Docker para cada nodo del sistema IoT	22
Figura 22 Implementación de artefactos de monitoreo del nodo Fog	23
Figura 23 Implementación de artefactos de monitoreo del nodo Edge	23
Figura 24 Implementación de artefactos de monitoreo del nodo Cloud	24
Figura 25 Datos monitoreados del ambiente de la casa en el nodo Edge.....	24
Figura 26 Datos recolectados por el Nodo Fog.....	25
Figura 27 Datos del Monóxido de Carbono de la cocina publicados en la Base de Datos	25
Figura 28 Datos de Temperatura y Humedad del dormitorio publicados en la Base de Datos.....	26
Figura 29 Datos de la carga de CPU publicados en la Base de Datos.....	26

Índice de Tablas

Tabla 1. Componentes de la aplicación supervisora del middleware	9
Tabla 2. Componentes de la Arquitectura del middleware - Nodos de Computación del Sistema IoT.....	12

RESUMEN

El desafío en el desarrollo de sistemas modernos (p. ej., IoT), radica en su complejidad y la constante evolución que experimentan, dado que los enfoques de desarrollo actuales se centran en la gestión del cambio durante la etapa de diseño, lo que implica una demanda sustancial de tiempo y esfuerzo por parte de los desarrolladores. En este contexto, el presente Trabajo de Titulación propone construir un prototipo de middleware basado en `models@run.time` para la implementación automática de artefactos de software en sistemas distribuidos, aprovechando el entorno de ejecución Node.js. Este middleware posee la capacidad de detectar de forma automática las modificaciones efectuadas en el modelo de alto nivel de arquitectura y crear los artefactos correspondientes, sin interrumpir el funcionamiento del sistema. La solución particularmente se aplica en un estudio de caso de monitorización en sistemas IoT con resultados muy favorables para afrontar la complejidad y dinamismo inherente a estos sistemas.

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

ABSTRACT

The challenge in developing modern systems (e.g., IoT) lies in their complexity and constant evolution, given that current development approaches focus on change management during the design stage, which involves a substantial demand of time and effort on the part of developers. In this context, this Thesis proposes to build a middleware prototype based on `models@run.time` for automatically implementing software artifacts in distributed systems, taking advantage of the Node.js execution environment. This middleware can automatically detect modifications made to the high-level architectural model and create the corresponding artifacts, without interrupting the system's operation. The solution is particularly applied in a case study of monitoring in IoT systems with very favorable results to address the complexity and dynamism inherent to these systems.

Keywords: Model-Driven Engineering (MDE), Internet of Things (IoT), Middleware, Runtime Models (`models@run.time`), Distributed Systems.



Brandon Iván Llivisaca Largo



Firmado electrónicamente por:
**LENIN XAVIER
ERAZO GARZON**

Ing. Lenín Xavier Erazo Garzón