UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

PLAN DE TRABAJO DE GRADO

FECHA DE PRESENTACIÓN: Bucaramanga, 16 de diciembre de 2022

TÍTULO: Mecanismos de adaptación autonómica de arquitectura software para la plataforma Smart Campus UIS

MODALIDAD: Trabajo de investigación

AUTOR: Daniel David Delgado Cervantes, 2182066

DIRECTOR: PhD. Gabriel Rodrigo Pedraza Ferreira, Escuela De Ingeniería De Sistemas e Informática

CODIRECTOR: MSc. Henry Andres Jimenez Herrera, Escuela De Ingeniería De Sistemas e Informática

ENTIDAD INTERESADA: Universidad Industrial de Santander

TABLA DE CONTENIDO

MECANISMOS DE ADAPTACIÓN AUTONÓMICA DE ARQUITECTURA SOFTWARE PARA LA PLATAFORMA SMART CAMPUS UIS

1 INTRODUCCIÓN

Dentro de la computación distribuida, una de las tendencias recientes dentro de la industria , es la búsqueda de maneras de reducir la complejidad de administrar los sistemas computacionales. A medida que estos crecen, en términos de tamaño y extensión, el costo humano de la administración de los sistemas se vuelve insustentable. En respuesta a esto, diferentes enfoques han surgido. Uno de estos es el *autonomic computing* (computación autonómica). Planteada originalmente por IBM en el año 2001, se presentaba como una posible solución a la problemática a partir de el diseño y la implementación de componentes auto-gestionados (Kephart, 2011).

Este acercamiento, aunque inicialmente concebido únicamente para sistemas distribuidos, puede también ser particularmente útil en otras ramas como lo son las

2 PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

La complejidad de los sistemas software ha ido en aumento. A medida que se hace la transición a arquitecturas orientadas a microservicios (Forrester Research, 2019); la computación distribuida es más común gracias a las soluciones *cloud* (Loukides, 2021) y la computación embebida se hace más presente; la administración y gestión de estos requiere de una mayor cantidad de recursos en términos técnicos y humanos con el fin de mantenerlos en los estados más óptimos respecto a los requerimientos del negocio. La búsqueda de reducir o abstraer la complejidad de la gerencia de estos sistemas se ha convertido en una necesidad (Lalanda, Diaconescu, y McCann, 2014).

Esta necesidad, así mismo, se presenta en los campos del Internet de las Cosas (IoT). Es en esta área de la computación embebida donde, debido a las cambiantes condiciones del mundo real, la arquitectura de estos sistemas de software se ve constantemente afectada. Una de las posibles soluciones se encuentra en la computación autonómica. Desde este enfoque, se tiene como objetivo sistemas con la capacidad de auto-gestión, es decir, sistemas con la capacidad de manejarse a ellos mismos dependiendo de las necesidades y las metas establecidas por los administradores del sistema (McCann y Huebscher, 2004).

Ahora, tenemos el caso de Smart Campus UIS, una plataforma de IoT de la Universidad Industrial de Santander. Esta ha realizado implementaciones parciales de una arquitectura autonómica con capacidad de auto-describirse (Jiménez, Cárcamo, y Pedraza, 2020). Dicho esto, y en búsqueda dar continuidad con los esfuerzos de desarrollo realizados en la plataforma, el siguiente paso a dar está en la implementación

de mecanismos de adaptación los cuales le concedan las propiedades de auto-configuración y auto-sanación.

3 OBJETIVOS

3.1 OBJETIVO GENERAL

- Implementar un conjunto de mecanismos autonómicos para permitir la adaptación de la Arquitectura Software IoT respecto a un modelo objetivo en la plataforma Smart Campus UIS

3.2 OBJETIVOS ESPECÍFICOS

- Proponer una notación (lenguaje) para describir una arquitectura objetivo de un sistema software IoT.
- Diseñar un mecanismo para determinar las diferencias existentes entre una arquitectura actual en ejecución y una arquitectura objetivo especificada.
- Diseñar un conjunto de mecanismos de adaptación que permitan disminuir las diferencias entre la arquitectura actual y la arquitectura objetivo.

4 MARCO DE REFERENCIA

4.1 COMPUTACIÓN AUTONÓMICA

4.1.1 MAPE-K

4.1.2 MECANÍSMOS DE ADAPTACIÓN

4.2 COMPUTACIÓN EMBEBIDA

4.2.1 INTERNET OF THINGS

4.3 LENGUAJES DE NOTACIÓN

4.3.1 SERIALIZACIÓN DE DATOS

4.3.2 GRAMÁTICA

4.4 ALGORITMIA DE COMPARACIÓN DE GRAFOS

5 METODOLOGÍA

Para el desarrollo del trabajo de grado, se ha definido un modelo de prototipado iterativo compuesto de 5 fases (Ver fig. 1). De esta manera, se avanzará a medida que se va completando la fase anterior y permitirá a futuro el poder iterar sobre lo que se ha desarrollado anteriormente.

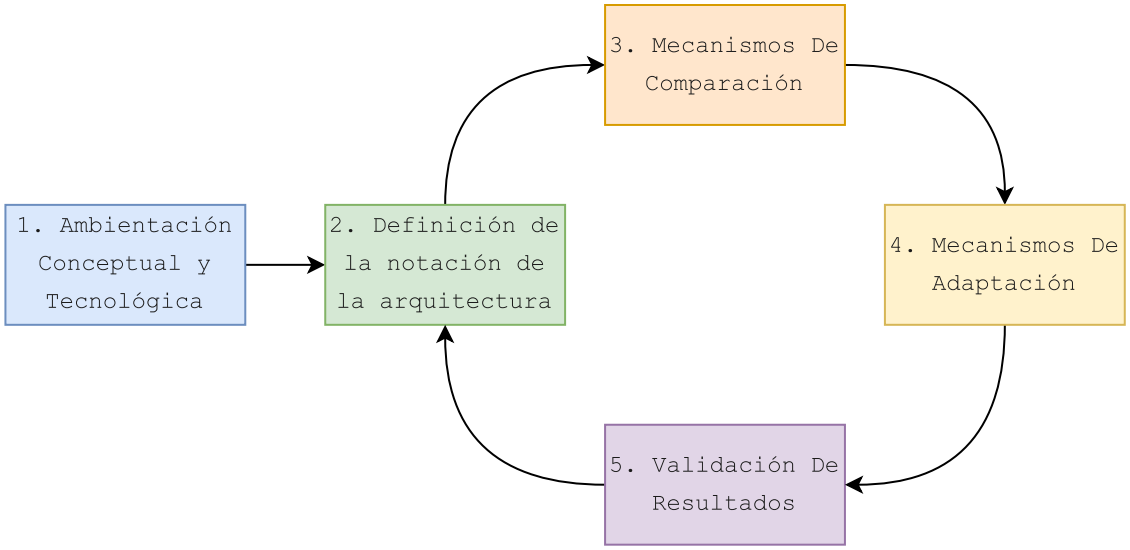

Figura 1: Metodología del proyecto

5.1 AMBIENTACIÓN CONCEPTUAL Y TECNOLÓGICA

La primera fase de la metodología se basa en la investigación de la literatura, al igual que de la industria, necesaria para cubrir las bases tanto conceptuales como técnicas necesarias para el desarrollo del proyecto.

ACTIVIDADES

5.1.1. Identificar las características principales de un sistema auto-adaptable.

5.1.2. Analizar los mecanismos de adaptación de la arquitectura.

5.1.3. Analizar los algoritmos empleados para la comparación de la comparación de las arquitecturas.

5.1.4. Establecer los criterios de selección para el lenguaje de notación.

5.1.5. Evaluar los posibles lenguajes de programación para la implementación a realizar.

5.1.6. Imprevistos.

5.1.7. Análisis, retroalimentación y conclusiones del desarrollo de la fase.

5.2 DEFINICIÓN DE LA NOTACIÓN DE LA ARQUITECTURA

La segunda fase está en la definición del como se realiza la declaración de la arquitectura. Partiendo de los criterios de selección establecidos el la fase 1, se espera determinar un lenguaje de notación el cual nos permita definir la arquitectura objetivo a alcanzar, al igual que la gramática correspondiente para poder realizar dicha declaración.

ACTIVIDADES

5.2.1. Seleccionar el lenguaje de marcado a usar a partir de los criterios establecidos.

5.2.2. Definir la gramática a usar para la definición de la arquitectura.

5.2.3. Implementar la traducción de la notación al modelo de grafos.

5.2.4. Determinar como se realizará la representación de los componentes y partes de la arquitectura.

5.2.5. Imprevistos.

5.2.6. Análisis, retroalimentación y conclusiones del desarrollo de la fase.

5.3 MECANISMOS DE COMPARACIÓN

Durante la tercera fase del proyecto, se buscará poder determinar e implementar cómo se realizará la comparación entre el estado de la arquitectura obtenido durante la auto-descripción de la misma y el objetivo establecido. Así mismo, y con el fin de reportar a los administradores de los sistemas, también será necesario definir *niveles* de similitud entre las 2 arquitecturas.

ACTIVIDADES

5.3.1. Seleccionar el mecanismos de comparación a usar para evaluación de estado de la arquitectura.

5.3.2. Implementar el mecanismo de comparación seleccionado.

5.3.3. Definir los diferentes niveles de similitud entre arquitecturas

5.3.4. Imprevistos.

5.3.5. Análisis, retroalimentación y conclusiones del desarrollo de la fase.

5.4 MECANISMOS DE ADAPTACIÓN

La cuarta fase del proyecto está orientada a la selección, al igual que la implementación, del conjunto de mecanismos de adaptación de la arquitectura.

5.4.1 ACTIVIDADES

- Actividad 1
- Actividad 2

5.5 VALIDACIÓN DE RESULTADOS

5.5.1 ACTIVIDADES

- Actividad 1
- Actividad 2

6 CRONOGRAMA

Se debe realizar un cronograma que relacione las actividades prioritarias del proyecto y el tiempo que destinará a cada una de ellas. Tenga en cuenta que el semestre tiene 16 semanas y debe desarrollar todo el trabajo de grado en este tiempo.

7 PRESUPUESTO

Descripción	Responsable	Valor	Cantidad	Precio
DIRECTOR DE PROYECTO PhD. Gabriel Rodrigo Pedraza Ferreira	UIS	COP 305.000/Hora	4 horas mensuales por 4 meses	COP 4’880.000

8 BIBLIOGRAFÍA

Forrester Research. (2019, Jun). *Mainframe in the age of cloud, ai, and blockchain.* Forrester Consulting. Descargado de https://www.ensono.com/resources/white-papers/old-workhorse-new-tech-mainframe-age-cloud-ai-and-blockchain-commissioned-study-conducted/

Jiménez, H., Cárcamo, E., y Pedraza, G. (2020). Extensible software platform for smart campus based on microservices. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, *2020*(E38), 270-282. Descargado de www.scopus.com

Kephart, J. (2011, 01). Autonomic computing: the first decade. En (p. 1-2). doi: 10.1145/1998582.1998584

Lalanda, P., Diaconescu, A., y McCann, J. A. (2014). *Autonomic computing: Principles, design and implementation*. Springer.

Loukides, M. (2021). *The cloud in 2021: Adoption continues*. Descargado de https://get.oreilly.com/rs/107-FMS-070/images/The-Cloud-in-2021-Adoption-Continues.pdf?mkt_tok=MTA3LUZNUy0wNzAAAAGISYNxeMWRA_a_GPKBEqQliGws2SImdqefJ4Ch11jEKmmSuN_ccGOOgoUv9enxj_0pbnchAdPjkL3QgDEdY4Xf5j_teuCKfiXTQIdg2jy7ETKmudbu

McCann, J. A., y Huebscher, M. C. (2004). Evaluation issues in autonomic computing. En H. Jin, Y. Pan, N. Xiao, y J. Sun (Eds.), *Grid and cooperative computing - gcc 2004 workshops* (pp. 597–608). Berlin, Heidelberg: Springer Berlin Heidelberg.