

**UNIVERSIDAD DE CONCEPCIÓN - CHILE  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

***SISTEMA HOLÓNICO DE MANUFACTURA PARA LA  
GESTIÓN DE SISTEMAS DE COLA EN AMBIENTES  
PRODUCTIVOS.***

por

**Italo Mauricio Montecinos Matamala**

Profesor Guía:

**Dr. Carlos E. Herrera L.**

Concepción, Mayo 2016

Tesis presentada a la

**DIRECCIÓN DE POSTGRADO  
UNIVERSIDAD DE CONCEPCIÓN**



Para optar al grado de

**MAGISTER EN INGENIERÍA INDUSTRIAL**



“Dedicado a mi familia y pareja  
que siempre me apoyaron y alentaron a seguir adelante”

# RESUMEN

## **SISTEMA HOLÓNICO DE MANUFACTURA PARA LA GESTIÓN DE SISTEMAS DE COLA EN AMBIENTES PRODUCTIVOS**

**“Italo Montecinos Matamala”**

**“Mayo 2016”**

**PROFESOR GUIA:** “Carlos E. Herrera L.”  
**PROGRAMA:** Magíster en Ingeniería Industrial

En la actualidad, la economía se ha globalizado, esto ha conllevado a que los mercados exijan productos de calidad y altamente personalizados, a un menor costo además, de una reducción del ciclo de vida de los mismos. Entonces, se impone una serie de propiedades a los nuevos sistemas de manufactura tales como: reconfiguración, flexibilidad, escalabilidad y robustez. Para hacer frente a las nuevas exigencias impuestas a los sistemas de manufactura, se han desarrollado nuevos paradigmas. Uno de estos, son los sistemas de control holónicos que si bien han sido probados en ambientes físicos y simulados, no han sido estudiados y adaptados para controlar un sistema productivo simulado a través de la teoría de redes de colas. En esta investigación, se propone una arquitectura de control holónica para la gestión de sistemas productivos simulados a través de la teoría de redes de colas. La arquitectura es testada en una plataforma compuesta por Matlab para simular el proceso y JADE para la ejecución de los holones. Los resultados demuestran que, en general, un sistema de control holónico tiene mejor performance que un control convencional controlando un sistema simulado a través de redes de colas en cuanto a utilización de máquinas, gestión de colas y cantidad de trabajos terminados.

**Palabras Claves:** **Sistemas de Manufactura, Sistema de Control Distribuido, Sistemas Holónicos de Manufactura, Redes de Cola.**

# ABSTRACT

## ***HOLONIC MANUFACTURING SYSTEM FOR QUEUEING SYSTEMS MANAGEMENT ON PRODUCTION ENVIROMENTS***

**“Italo Montecinos Matamala”**

**“Mayo 2016”**

**THESIS SUPERVISOR:** “Carlos E. Herrera L.”  
**PROGRAM:** Master in Industrial Engineering

At present, the economy has become globalized, this has led to markets require quality products, highly personalized at lower cost and also a reduction of product life cycle. This has imposed a number of properties to new manufacturing systems such as: reconfiguration, flexibility, scalability and robustness. Example of one of them is the holonic control systems, although they have been tested in physical and simulated environments, they have not been studied and adapted for controlling a production system simulated through queueing networks theory. In this research, architecture holonic management control of production systems based on queueing is proposed. The results show that, in general, holonic system control has better performance than a conventional control controlling a simulated process through queueing networks theory regarding machines utilization, queueing management and finished jobs quantity.

**Keywords: Manufacturing Systems, Distributed Control Systems, Holonic Manufacturing Systems, Queueing Networks.**

# ÍNDICE

## CAPÍTULO 1

<b>Introducción .....</b>	<b>1</b>
1.1 Hipótesis de Trabajo .....	3
1.2 Objetivo General .....	3
1.3 Objetivos Específicos .....	3
1.4 Estructura de Tesis .....	4

## CAPÍTULO 2

<b>Estado del Arte.....</b>	<b>5</b>
2.1 Introducción al Capítulo.....	5
2.2 La Industria Manufacturera y su relevancia para la Economía Mundial .....	5
2.3 Sistemas de Control de Manufactura Tradicionales.....	8
2.4 Justificación para la evolución los Sistemas de Manufactura .....	12
2.5 Sistemas de Control de Manufactura .....	14
2.6 Enfoques y paradigmas emergentes para el Sistemas de Control de Manufactura....	17
2.7 Aplicaciones de Sistemas Multi-agentes y Holónicos .....	29
2.8 Simulación basada en Agentes.....	31
2.9 Brecha detectada .....	33

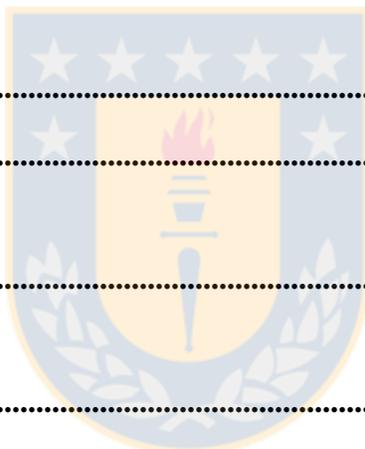
## CAPÍTULO 3

<b>Materiales y Métodos.....</b>	<b>34</b>
3.1 Introducción al capítulo.....	34
3.2 Arquitectura propuesta .....	34
3.3 Método de Experimentación .....	40

## CAPÍTULO 4

<b>Resultados.....</b>	<b>56</b>
4.1 Introducción al capítulo.....	56
4.2 Escenario base: Control Convencional sin Falla.....	56

4.3	Escenario 1: Falla en el servidor $S_{1I}$ , sin recuperación .....	57
4.4	Escenario 2: Falla en el servidor $S_{1I}$ , sin recuperación .....	60
4.5	Escenario 3: Falla en el servidor $S_{2I}$ , sin recuperación .....	63
4.6	Escenario 4: Falla en el servidor $S_{2I}$ , sin recuperación .....	66
4.7	Escenario 5: Falla en el servidor $S_{3I}$ , sin recuperación .....	69
4.8	Escenario 6: Falla en el servidor $S_{3I}$ , sin recuperación .....	72
4.9	Escenario 7: Falla en el servidor $S_{1I}$ , con recuperación .....	75
4.10	Escenario 8: Falla en el servidor $S_{2I}$ , con recuperación .....	78
4.11	Escenario 9: Falla en el servidor $S_{3I}$ , con recuperación .....	81
4.12	Comentarios generales .....	84
<b>CAPÍTULO 5</b>		
	<b>Conclusiones</b> .....	<b>85</b>
	<b>REFERENCIAS</b> .....	<b>89</b>
<b>ANEXO 1</b>		
	<b>Plataforma de Desarrollo</b> .....	<b>94</b>
<b>ANEXO 2</b>		
	<b>Códigos Fuente</b> .....	<b>102</b>
<b>ANEXO 3</b>		
	<b>Gráficos de Resultados</b> .....	<b>125</b>



## ÍNDICE DE FIGURAS

Figura 1: Valor Agregado Mundial producido por la Industria Manufacturera 2005-2013.....	5
Figura 2: Valor Agregado aportado por la Industria Manufacturera de la OECD y UE. 2014 ..	6
Figura 3: Valor Agregado de la industria manufacturera de las tres mayores potencias mundiales 2005-2013.....	6
Figura 4: Aporte al Valor Agregado mundial de Países Sudamericanos. Fuente: Elaboración propia basada en información del Banco Mundial. ....	7
Figura 5: Valor Agregado de la Industria Manufacturera en Sudamérica 2005-2013. Fuente: Elaboración propia basada en información del Banco Mundial. ....	7
Figura 6: Arquitectura CIM según ISA-95 y su implementación industrial. Fuente: Imagen adaptada del Estándar ANSI/ISA-95. ....	9
Figura 7: Evolución de MRP a ERP. Fuente: Imagen adaptada de [18]. ....	10
Figura 8: Resumen de Propiedades a exhibir por el sistema de manufactura y requisitos para su sistema de control .....	16
Figura 9: Clasificación de arquitecturas de control y sus implicancias. Imagen adaptada de [17].....	18
Figura 10: a) Diseño de Sistema de Control basado en VSM. b) Estructura de un producto Inteligente usando VSM Imagen adaptada de [5].....	21
Figura 11: Arquitectura de un Holón. Imagen adaptada de Fuente especificada no válida.. ...	23
Figura 12: Arquitectura de Referencia PROSA Fuente: Imagen adaptada de [29]. ....	26
Figura 13: Reorganización Dinámica de ADACOR. Imagen adaptada de [11].....	27
Figura 14: Evolución de arquitecturas y aplicaciones basadas en agentes y su aplicación en la industria. Fuente: Imagen extraída de [6] .....	29
Figura 15: Etapas de transición de arquitecturas. Fuente: Elaboración propia.....	35
Figura 16: Red de Cola aplicable al algoritmo de control propuesto. Fuente: Elaboración propia. ....	37
Figura 17: a) Diagrama de Flujo Holón Ruteo (HR). b) Diagrama de Flujo Holón de Línea (HL). Fuente: Elaboración propia. ....	38
Figura 18: Proceso de experimentación.....	41
Figura 19: Holarquía en Arquitectura Centralizada (Proceso sin Falla).....	43
Figura 20: Holarquía en Arquitectura Descentralizada-Heterárquica (Proceso con Falla) .....	43
Figura 21: Proceso Implementado en Simulink.....	48

Figura 22: Proceso Controlado y Supervisado por CRC-H.....	50
Figura 23: Diagrama de Flujo del algoritmo de control para Holón de Ruteo y Holón de Línea .....	52
Figura 24: Layout del GUI diseñado para el Holón de Ruteo .....	53
Figura 25: Layout del GUI diseñado para el Holón de Línea.....	54
Figura 26: Resumen de gráficas línea 1, servidores 1 y 2, Escenario 1.....	59
Figura 27: Trabajos terminados en línea 1 y 2, Escenario 1 .....	59
Figura 28: Resumen de gráficas línea 1, servidores 1 y 2, Escenario 2.....	62
Figura 29: Trabajos terminados en línea 1 y 2, Escenario 2.....	62
Figura 30: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 3.....	65
Figura 31: Trabajos terminados en línea 2 y 3, Escenario 3.....	65
Figura 32: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 4.....	68
Figura 33: Trabajos terminados en línea 2, Escenario 4.....	68
Figura 34: Resumen de gráficas línea 3, servidores 1 y 2, Escenario 5.....	71
Figura 35: Trabajos terminados en línea 2 y 3, Escenario 5.....	71
Figura 36: Resumen de gráficas línea 3, servidores 1 y 2, Escenario 6.....	74
Figura 37: Trabajos terminados en línea 2 y 3, Escenario 6.....	74
Figura 38: Resumen de gráficas línea 3, servidores 1 y 2, Escenario 7.....	77
Figura 39: Trabajos terminados en línea 2 y 3, Escenario 7 .....	77
Figura 40: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 8.....	80
Figura 41: Trabajos terminados en línea 2 y 3, Escenario 8.....	80
Figura 42: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 9.....	83
Figura 43: Trabajos terminados en línea 2 y 3, Escenario 9.....	83
Figura 44: Expansión del sistema de control holónico propuesto. ....	87
Figura 45: Comunicación Simulink/JADE, imagen adaptada de [39].....	96
Figura 46: Interfaz gráfica MACSimJX .....	99
Figura 47: Interfaz de JADE ejecutando el Agent Coordinator de MACSimJX.....	100
Figura 48: Interfaz Gráfica JADE con MACSimJX y Holones de aplicación en ejecución ..	100
Figura 49: Holones de Línea y Ruteo en operación y comunicados con Simulink .....	101
Figura 50: Gráficos Línea 0 / Servidores 1 y 2, Control Convencional sin falla.....	125
Figura 51: Gráficos Línea 1 / Servidores 1 y 2, Control Convencional sin falla.....	126
Figura 52: Gráficos Línea 2 / Servidores 1 y 2, Control Convencional sin falla.....	126
Figura 53: Gráficos Línea 3 / Servidores 1 y 2, Control Convencional sin falla.....	127

Figura 54: Trabajos despachados y terminados en L1, L2 y L3, Control Convencional sin falla .....	127
Figura 55: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 1 .....	128
Figura 56: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 1 .....	128
Figura 57: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 1 .....	129
Figura 58: Trabajos despachados y Terminados en L1, L2 y L3, Escenario 1 .....	129
Figura 59: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 2 .....	130
Figura 60: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 2 .....	130
Figura 61: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 2 .....	131
Figura 62: Trabajos despachados y terminados en L1, L2 y L3, Escenario 2 .....	131
Figura 63: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 3 .....	132
Figura 64: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 3 .....	132
Figura 65: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 3 .....	133
Figura 66: Trabajos despachados y terminados en L1, L2 y L3, Escenario 3 .....	133
Figura 67: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 4 .....	134
Figura 68: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 4 .....	134
Figura 69: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 4 .....	135
Figura 70: Trabajos despachados y terminados en L1, L2 y L3, Escenario 4 .....	135
Figura 71: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 5 .....	136
Figura 72: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 5 .....	136
Figura 73: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 5 .....	137
Figura 74: Trabajos despachados y terminados en L1, L2 y L3, Escenario 5 .....	137
Figura 75: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 6 .....	138
Figura 76: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 6 .....	138
Figura 77: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 6 .....	139
Figura 78: Trabajos despachados y terminados en L1, L2 y L3, Escenario 6 .....	139
Figura 79: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 7 .....	140
Figura 80: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 7 .....	140
Figura 81: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 7 .....	141
Figura 82: Trabajos despachados y terminados en L1, L2 y L3, Escenario 7 .....	141
Figura 83: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 8 .....	142
Figura 84: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 8 .....	142
Figura 85: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 8 .....	143

Figura 86: Trabajos despachados y terminados en L1, L2 y L3, Escenario 8.....	143
Figura 87: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 9 .....	144
Figura 88: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 9 .....	144
Figura 89: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 9 .....	145
Figura 90: Trabajos despachados y terminados en L1, L2 y L3, Escenario 9 .....	145



## ÍNDICE DE TABLAS

Tabla 1: Tabla comparativa control convencional vs Control Distribuido [1] .....	19
Tabla 2: Aplicación de sistemas MAS y HMS según ámbito de aplicación .....	30
Tabla 3: Plataformas de Ejecución de Sistemas basados en Agentes .....	31
Tabla 4: Descripción de Holones que componen el sistema de control propuesto.....	36
Tabla 5: Parámetro de red de cola a simular.....	40
Tabla 6: Definición de Holones .....	42
Tabla 7: Software necesario para la experimentación .....	45
Tabla 8: Hardware necesario para la experimentación .....	46
Tabla 9: Bloques del Toolbox Simevents para la aplicación.....	47
Tabla 10: Interfaz Entrada/Salida entre Simulink/JADE.....	49
Tabla 11: Regla de asignación de probabilidades de HR .....	52
Tabla 12: Definición de escenarios de experimentación .....	55
Tabla 13: Resultados de Métricas Escenario Base .....	56
Tabla 14: Parámetros del escenario 1 .....	57
Tabla 15: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 1 .....	58
Tabla 16: Comparación porcentual entre controles, Escenario 1 .....	59
Tabla 17: Parámetros del escenario 2 .....	60
Tabla 18: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 2 .....	61
Tabla 19: Comparación porcentual entre controles, Escenario 2 .....	62
Tabla 20: Parámetros del escenario 3 .....	63
Tabla 21: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 3 .....	64
Tabla 22: Comparación porcentual entre controles, Escenario 3 .....	65
Tabla 23: Parámetros del escenario 4 .....	66
Tabla 24: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 4 .....	67
Tabla 25: Comparación porcentual entre controles, Escenario 4 .....	68
Tabla 26: Parámetros del escenario 5 .....	69

Tabla 27: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 5 .....	70
Tabla 28: Comparación porcentual entre controles, Escenario 5 .....	71
Tabla 29: Parámetros del escenario 6 .....	72
Tabla 30: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 6 .....	73
Tabla 31: Comparación porcentual entre controles, Escenario 6 .....	74
Tabla 32: Parámetros del escenario 7 .....	75
Tabla 33: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 7 .....	76
Tabla 34: Comparación porcentual entre controles, Escenario 7 .....	77
Tabla 35: Parámetros del escenario 8 .....	78
Tabla 36: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 8 .....	79
Tabla 37: Comparación porcentual entre controles, Escenario 8 .....	80
Tabla 38: Parámetros del escenario 9 .....	81
Tabla 39: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 9 .....	82
Tabla 40: Comparación porcentual entre controles, Escenario 9 .....	83
Tabla 41: Tabla resumen Métricas de comparación porcentual Control CRC-H vs Convencional .....	84

# CAPÍTULO 1

## Introducción

El desarrollo de los sistemas de manufactura ha sido constante y necesario en el tiempo producto factores del tipo Económico y Operacional. La economía global exige productos de mayor calidad, altamente personalizados a un menor costo, además, de una reducción de los tiempos de mercados y los ciclos de vida de los productos, mientras que, los factores operaciones también han impactado en la evolución de los sistemas. Estos factores son: retardos en la toma de decisiones producto de la jerarquización de las decisiones y los problemas propios a nivel de piso y de gestión de la planta tales como falla de máquinas, ausentismo laboral, cambios de pedido urgentes, comportamiento cambiante en clientes y retraso de operadores. Ambos factores han llevado a concluir a la comunidad científica que es necesario mejorar la respuesta dinámica de los sistemas de control de manufactura ante una emergencia o algún imprevisto [1, 2, 3].

Los sistemas de control tradicionales los cuales son de tipos centralizados y jerárquicos, permiten una optimización global del sistema pero no responden con eficiencia y eficacia ante cambios dinámicos en los procesos de manufactura producto de la rigidez de su arquitectura [4]. Dado lo anterior, se ha realizado en la comunidad científica extensos estudios que tienen como denominador común el desarrollo de sistemas de manufactura que exhiban propiedades tales como: inteligencia, robustez, adaptación a perturbaciones, reconfiguración, escalabilidad y flexibilidad. Esto conlleva a nuevos requerimientos para los sistemas de control de manufactura como: contar con una arquitectura descentralizada, toma de decisiones de control global en forma conjunta, ser auto-organizados, reactivos, proactivos y considerar entidades que interactúen de forma generalizada y flexible.

Producto de estos nuevos requerimientos, se han desarrollado nuevos sistemas de control con arquitecturas con distintos enfoques tales como: semi-jerárquica, heterárquica e híbrida (arquitectura dinámica). Estos nuevos enfoques son utilizados en conjunto con nuevos paradigmas que se adaptan a estas arquitecturas como por ejemplo: Modelo de Sistemas Viabiles (VSM) [5], Multi-agentes (MAS) [6] y Sistemas Holónicos de Manufactura (HMS) [7].

Respecto a las arquitecturas heterárquicas, una línea de investigación que ha cobrado fuerza son los sistemas de control basados en productos inteligentes (Product Driven Control Systems, PDCS). En este enfoque la inteligencia y la autonomía se traen lo más cerca posible del sistema físico, o incluso embebida en el mismo producto. Esto permite a las entidades de decisión trabajar y colaborar juntas con el objetivo de reaccionar rápida y autónomamente, en vez de solicitar decisiones de control a un nivel más alto de decisión el cual genera retardos de tiempos de respuesta. Algunos trabajos desarrollados con este enfoque pueden ser revisados en [8], [5], [9] y [10]. Estudios [9] revelan que si bien las arquitecturas totalmente heterárquicas evidencian ser sistemas altamente reactivos ante decisiones locales, a la vez, evidencian que cada entidad decisional en la arquitectura presenta dificultad para predecir su propio comportamiento en el futuro y el de las demás entidades, conllevando a la obtención de soluciones no óptimas a mediano y largo plazo, este síntoma es denominado Miopía [8].

Las arquitecturas híbridas son un enfoque ampliamente estudiado que principalmente fueron desarrollados para mitigar el síntoma de la miopía que presentan las arquitecturas heterárquicas. La arquitectura se comporta de forma centralizada cuando el sistema está estable y de forma descentralizada cuando el sistema es perturbado. Una de las principales arquitecturas que considera una auto-organización dinámica es ADACOR [11], esta propuesta, basada en la arquitectura holónica de referencia PROSA, realiza un enfoque centralizado cuando el objetivo es la optimización y un enfoque descentralizado cuando existe un evento inesperado. Los resultados obtenidos en la aplicación de este enfoque, demuestra que las arquitecturas híbridas son más eficientes que las arquitecturas estáticas ya que aprovechan las ventajas de optimización global de una arquitectura jerárquica y la reactividad de una arquitectura heterárquica cuando existe una perturbación en el sistema [8] [11].

Específicamente respecto a los sistemas de control holónicos, si bien estos han sido probados tanto en ambientes físicos como simulados [6] [12], no existen investigaciones que apliquen las arquitecturas desarrolladas en un ambiente de simulación utilizando la teoría de redes de colas para representar un sistema productivo con el fin de evaluar y validar la performance de los sistemas de control de forma experimental.

Justificado por lo anterior, este trabajo busca desarrollar un sistema de control con arquitectura híbrida, basada en el paradigma de holónico para controlar un proceso industrial

simulado a través de la teoría de colas y que tenga la habilidad de reaccionar eficientemente ante la presencia de perturbaciones.

### **1.1 Hipótesis de Trabajo**

Un sistema de control holónico aplicado para el control de un proceso simulado utilizando la teoría de redes de cola proporciona mayor robustez ante cambios dinámicos y perturbaciones que un Sistema de Control Convencional Centralizado.

### **1.2 Objetivo General**

Desarrollar y evaluar un sistema holónico para la gestión de un sistema productivo simulado a través de la teoría de redes de colas.

### **1.3 Objetivos Específicos**

- ✓ Estudiar trabajos relacionados con Sistemas Holónicos de manufactura y trabajos provenientes de sistemas inteligentes de manufactura.
- ✓ Estudiar modelos provenientes de la teoría de colas, específicamente redes de Jackson.
- ✓ Proponer una arquitectura de control holónica para gestión de trabajos de un proceso simulado a través de redes Jackson.
- ✓ Desarrollar una plataforma de simulación de sistemas de control holónico estándar, adaptable y mejorable en el tiempo para probar nuevos enfoques basados en el paradigma holónico de forma sustentable.
- ✓ Comparar los resultados del sistema de control propuesto respecto de un control convencional.

## 1.4 Estructura de Tesis

El presente trabajo está estructurado de la siguiente forma:

En el Capítulo 2, se presenta la revisión bibliográfica, donde se analizan distintas investigaciones asociadas a los sistemas de control utilizados hasta el momento para manejar procesos productivos, para finalmente plantear la dirección que tomará el presente estudio.

En el Capítulo 3, se presenta un nuevo sistema de control de manufactura, una plataforma de simulación y se define el procedimiento de experimentación para validación de la arquitectura propuesta.

En el Capítulo 4, se discuten los resultados de los experimentos realizados.

En el Capítulo 5, se presentan las conclusiones obtenidas de acuerdo a los resultados experimentales y sugerencias para realizar trabajos futuros.



# CAPÍTULO 2

## Estado del Arte

### 2.1 Introducción al Capítulo

El presente capítulo tiene como objetivo analizar los sistemas de control tradicionalmente utilizados en los sistemas de manufactura, así como las principales líneas de investigación para los nuevos enfoques. Finalmente se plantea la dirección que tomará el presente estudio en función de la brecha de conocimiento detectada.

### 2.2 La Industria Manufacturera y su relevancia para la Economía Mundial

La industria manufacturera a nivel mundial representa actualmente uno de los sectores con mayor generación de ingresos económicos. A continuación se detalla en un contexto mundial y nacional el valor agregado<sup>1</sup> y el % del Producto Interno Bruto (PIB) aportado por el sector manufacturero según la última información disponible del Banco Mundial<sup>2</sup> [13] :

En el contexto mundial, el valor agregado aportado por la industria manufacturera ha sido creciente en el tiempo llegando el 2013 a US\$11.92 trillones<sup>3</sup> (ver Figura 1), con un aporte al PIB mundial de 15.8%.

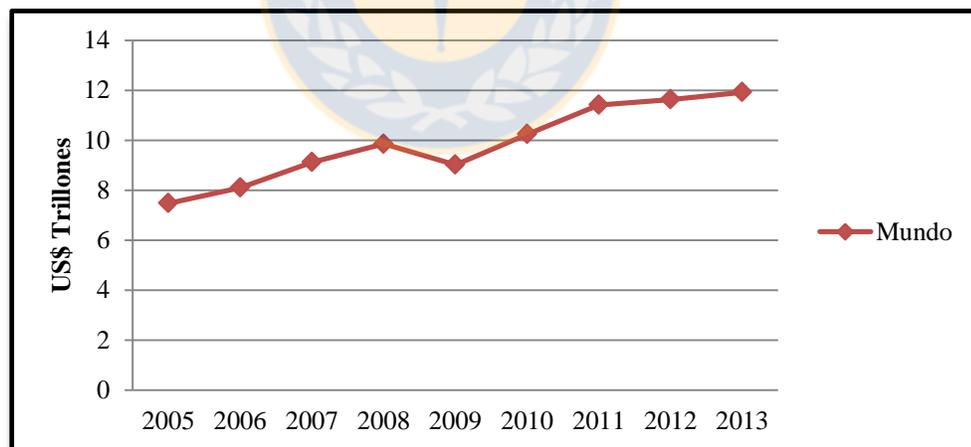


Figura 1: Valor Agregado Mundial producido por la Industria Manufacturera 2005-2013

<sup>1</sup> El valor agregado es la producción neta de un sector después de sumar todos los productos y restar los insumos intermedios.

<sup>2</sup> Los datos analizados de valor agregado y PIB disponibles en el banco mundial corresponden a las industrias manufactureras pertenecientes a las divisiones 15 a 37 de la Clasificación Internacional Industrial Uniforme (CIIU).

<sup>3</sup> La definición de trillones es según la lista corta, es decir, 1 Trillón=10<sup>12</sup>.

Por otro lado, los países miembros de la Unión Europea (UE) generan US\$2.4 trillones correspondientes al 20% del valor agregado mundial y la Organización para la Cooperación y el Desarrollo Económico (OCDE) alcanza un valor agregado de US\$6,4 trillones correspondientes al 53.7% del PIB mundial (ver Figura 2).

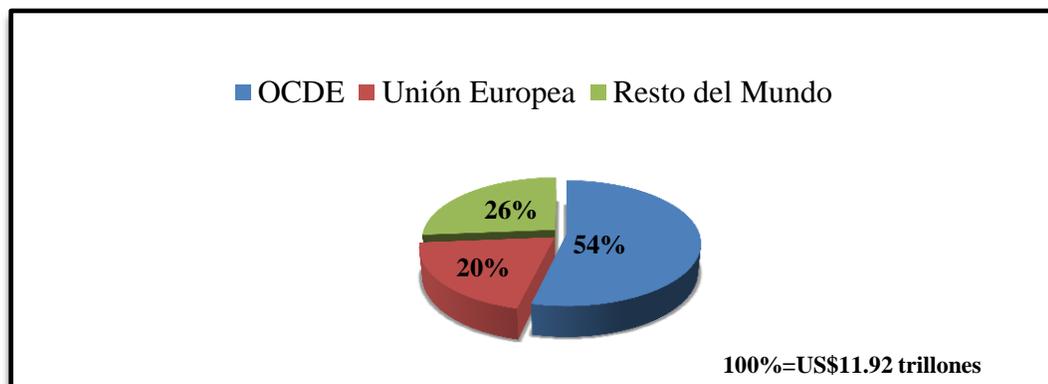


Figura 2: Valor Agregado aportado por la Industria Manufacturera de la OECD y UE. 2014

Quien lidera la lista de países OCDE con mayor incidencia en estos índices es la República de China con un valor agregado de US\$2,9 trillones, le sigue Estados Unidos con un VA de US\$1,9 trillones y en tercer lugar Japón con un valor agregado de US\$0,9 trillones (Ver Figura 3). Por otro lado, un reporte publicado por la National Association Of Manufacturers [14], indica que el sector de manufactura en Estados Unidos, genera aproximadamente 17.6 millones de empleados, mientras que un estudio elaborado por la Comisión Europea [15], declara que el 70% de los empleos generados en la UE tiene relación con la manufactura, mientras otro estudio generado por la misma entidad [16] declara que la manufactura da trabajo a 34 millones de personas en el año 2006 a toda Europa con 230.000 empresas creadas de veinte o más empleados. Con los índices indicado anteriormente se puede deducir la importancia de la IM alrededor de todo el mundo generando.

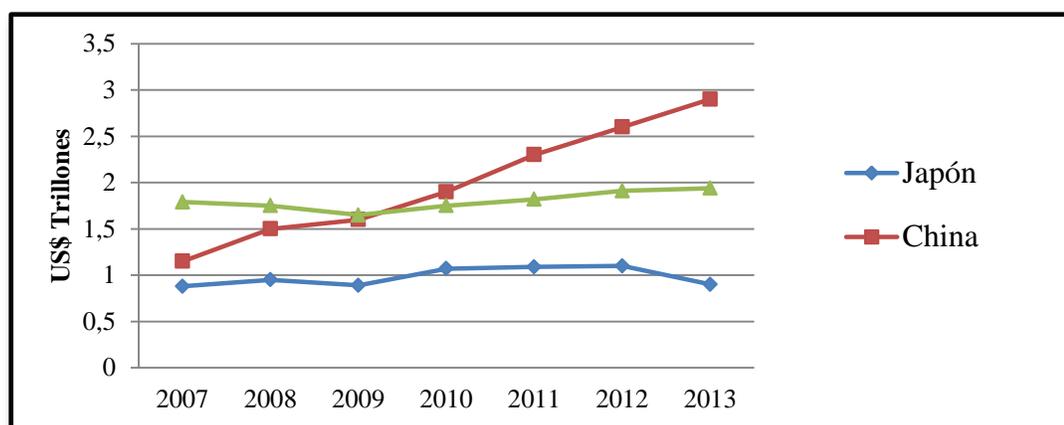
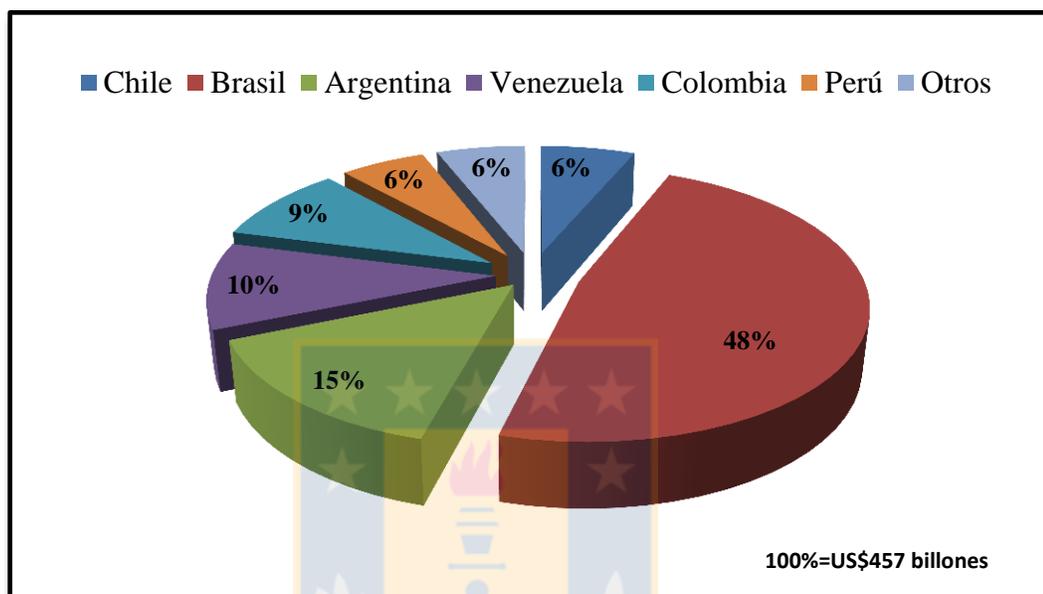
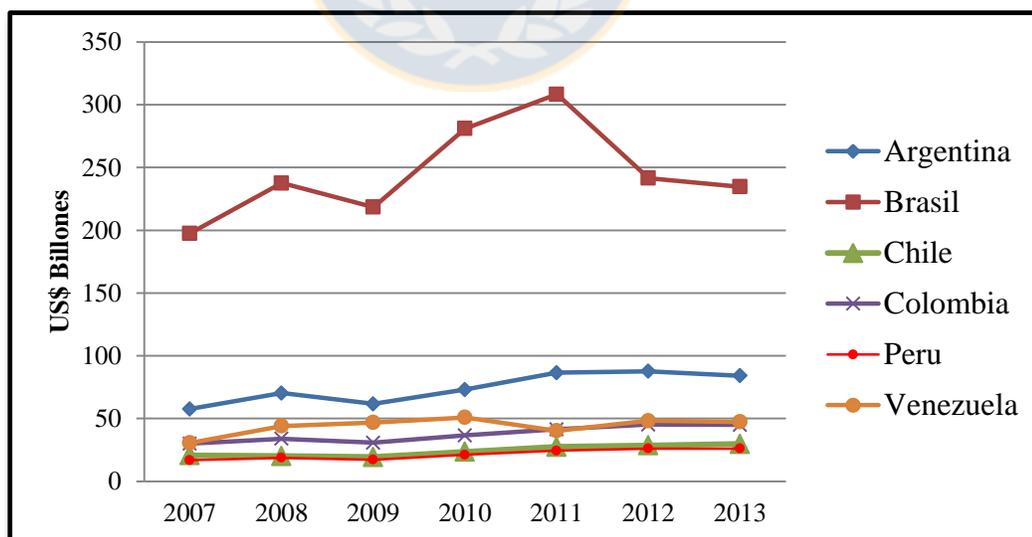


Figura 3: Valor Agregado de la industria manufacturera de las tres mayores potencias mundiales 2005-2013. Fuente: elaboración propia basada en información del Banco Mundial.

En el contexto nacional, Chile se ubica en el lugar N°44 a nivel mundial y 5to a nivel sudamericano con un valor agregado de US\$29 billones<sup>4</sup>, lo cual representa un 6% del total del valor agregado aportado por Sudamérica (ver Figura 4) y un aporte al PIB del país de 12.4% en el año 2014. Con estas cifras, Chile se encuentra muy por debajo de Brasil quien cuenta con un valor agregado de US\$218.8 billones, pero por sobre Perú, Uruguay, Paraguay, Bolivia y Ecuador (ver Figura 5).



**Figura 4: Aporte al Valor Agregado mundial de Países Sudamericanos.**  
Fuente: Elaboración propia basada en información del Banco Mundial.



**Figura 5: Valor Agregado de la Industria Manufacturera en Sudamérica 2005-2013.**  
Fuente: Elaboración propia basada en información del Banco Mundial.

<sup>4</sup> La definición de billones es según la lista corta, es decir, 1 Billón=10<sup>9</sup>.

### 2.3 Sistemas de Control de Manufactura Tradicionales

Los paradigmas de manufactura permanentemente han ido evolucionando en función de las demandas del cliente, tecnología disponible y las tendencias económicas. Con la primera revolución industrial, las máquinas fueron introducidas a la producción y con ello un aumento en la productividad. A comienzos del siglo 20 Henry Ford comienza a construir una cantidad limitada de automóviles personalizados por los clientes, posteriormente introduce el concepto de producción en masa en el cual consiste en la producción de un solo producto en grandes cantidades utilizando líneas de ensamblaje rígidas. El ciclo de tarea (cantidad de tiempo que el operador trabaja antes de repetir las mismas operaciones) se disminuyó de 514 minutos a 1,2 minutos [17].

Posteriormente, en la década de los 70 y 80, la globalización de la economía, el poder creciente de los clientes y la heterogeneidad de los mercados, produjeron que el paradigma de producción en masa ya no fuera eficiente. La rigidez del modelo comenzó a ser un problema para la competitividad de la empresa y entonces, la producción en masa comenzó a ser viable solo para algunos productos. Las empresas comenzaron a desarrollar paradigmas para lograr flexibilidad en la producción y en los años 90 el objetivo era el incremento de la agilidad. El concepto de agilidad se refiere a la capacidad de responder rápida y rentablemente a cambios inesperados y se basa en cuatro lineamientos: entregar valor al cliente, capacidad de reaccionar a los cambios, el valor de los conocimientos y habilidades humanas, y la capacidad para constituir asociaciones virtuales [17].

Hoy en día, la producción está basada en demandas específicas del cliente [6], [17], [18], y es entonces donde se nace el paradigma de personalización en masa el cual apunta a satisfacer necesidades individuales de clientes. La personalización en masa requiere que el sistema de manufactura sea flexible y ágil de manera de producir múltiples variaciones de un producto personalizable al precio de un producto estándar.

En la práctica, durante los años 70 y 80, las empresas creaban islas de procesos automatizados para resolver problemas específicos, esta solución conllevaba a una redundancia de información y desaprovechamiento de recursos. Con el fin de integrar estas islas durante en los años 80, nace el paradigma Manufactura Asistida por Computador (Computer Integrated Manufacturing, CIM), el cual consiste en la integración de las actividades empresariales con la producción utilizando herramientas tecnológicas como bases de datos y redes informáticas industriales. Este paradigma es totalmente jerárquico, basado

jerarquizando los niveles de decisión en rangos de tiempo decrecientes, es decir, decisiones de tipo estratégicas, tácticas y operacionales asignando entidades de decisión dependientes entre sí con una optimabilidad a largo plazo (optimización global), mientras que a la vez soporta una optimización a corto plazo, es decir, agilidad y reactividad [17].

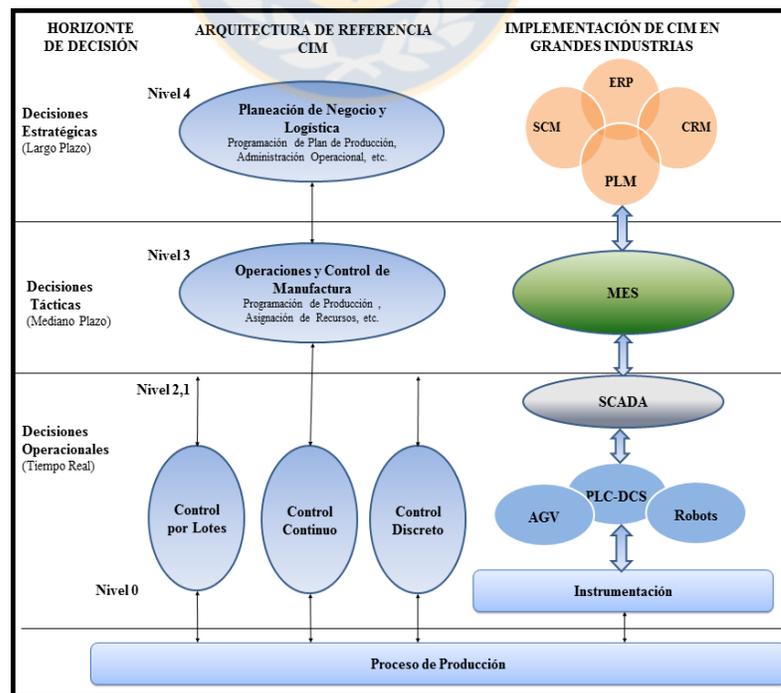
Las principales ventajas del paradigma CIM son las siguientes:

1. Incrementa la productividad.
2. Incrementa la flexibilidad.
3. Incrementa la calidad de los productos.
4. Reduce los tiempos de diseño.
5. Reduce el trabajo en progreso.

El paradigma CIM es conocido por proporcionar soluciones cercanas a las óptimas cuando se cumplan algunos supuestos como por ejemplo [17]:

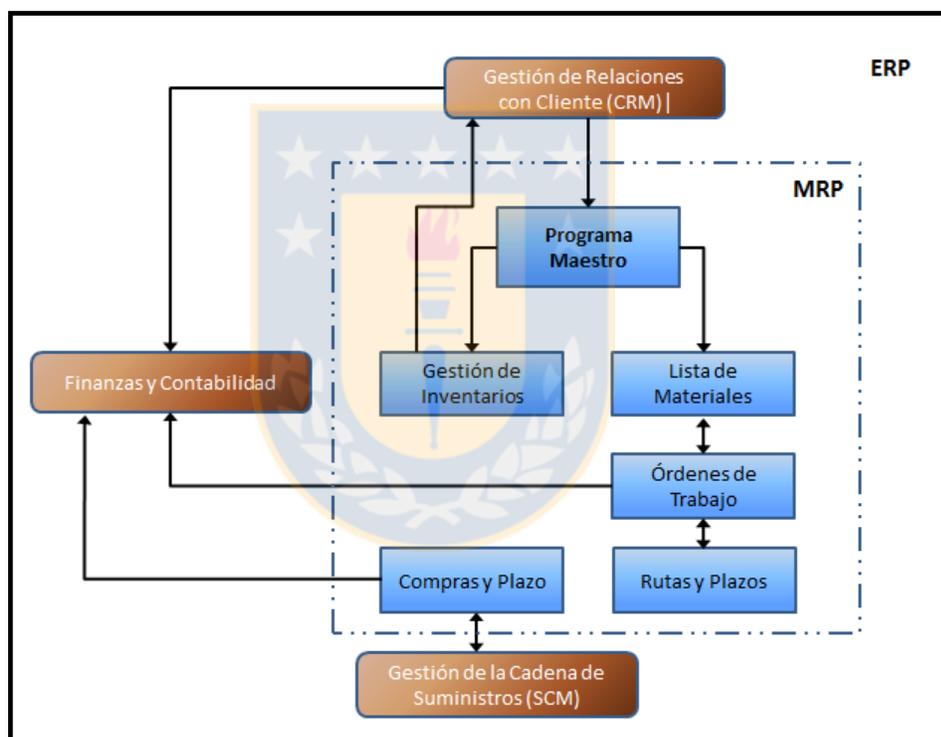
1. Disponibilidad y fiabilidad del suministro a largo plazo y la demanda.
2. Comportamiento óptimo y alta fiabilidad de sistemas de producción.
3. Baja diversidad de productos.
4. Observabilidad y capacidad de control de todas las posibles variables internas.

A continuación, se describe la arquitectura CIM según el estándar ISA-95, además, la Figura 6 se puede observar su implementación en la industria:



**Figura 6: Arquitectura CIM según ISA-95 y su implementación industrial.**  
Fuente: Imagen adaptada del Estándar ANSI/ISA-95.

**Nivel 4 – Decisiones Estratégicas:** asocia todas las actividades que soportan las decisiones de largo plazo, tales como: relaciones con el cliente (Customer relationship management, CRM), cadena de suministro SCM (Supply chain management, SCM), administración del ciclo de vida de los productos (Product lifecycle management, PLM) plan básico de producción y las finanzas/contabilidad. En la práctica, estos sistemas evolucionaron desde los años 70' con la Planificación de Requerimientos de Materiales (Material Requirements Planning, MRP) a la Planificación de Recursos de Manufactura (Manufacturing Resource Planning, MRP2) en los años 80' y a la más reciente en los años 90' Planificación de Recursos Empresariales (Enterprise Resource Planning, ERP), el cual integra Finanzas/Contabilidad, SCM y CRM. La evolución de los componentes desde un MRP a ERP se resume en la Figura 7.



**Figura 7: Evolución de MRP a ERP.**  
Fuente: Imagen adaptada de [18].

**Nivel 3 – Decisiones Tácticas:** corresponde a las actividades empresariales que soportan las decisiones tácticas de mediano-corto plazo, tales como: Presentación de informes sobre la producción, incluyendo los costes variables de fabricación, recolección y mantención de datos de inventario, fuerza de trabajo, materias prima y uso de energía, interpretación y análisis de datos, programación y asignación de recursos, optimización de funciones de control y costos. En la práctica, el nivel 3 se implementa a través de una plataforma denominada Sistema de

Ejecución de Manufactura (Manufacturing execution system, MES), el cual proporciona la información necesaria para optimizar los sistemas productivos desde la entrada de una orden de producción hasta el producto terminado. Este sistema enlaza la información de Planificación proveniente del Nivel 4 con los sistemas de control en piso de la fábrica (Nivel 2).

**Nivel 2, 1 y 0 – Decisiones Operaciones:** es el conjunto de todas las actividades de corto plazo. Este nivel define la célula o funciones de supervisión, operación y control de producción. Existen diferentes formas de implementar este nivel que depende del proceso y de la estrategia de producción, A continuación, se ejemplifica una implementación en la industria:

- Nivel 2: un sistema de control y adquisición de datos (SCADA), incluido Controladores Lógicos Programables (PLC), que permite a operadores en salas de control monitorear y controlar el proceso en tiempo real.
- Nivel 1: instrumentación convencional, analítica, vehículos autónomos (AGV), robots, etc.
- Nivel 0: proceso físico.

## 2.4 Justificación para la evolución los Sistemas de Manufactura

El incentivo para la evolución de los sistemas de manufactura centralizados a un sistema de manufactura distribuido se produce por dos factores: Económico y Operacional.

Factor Económico: La economía se ha globalizado, y los mercados exigen productos de calidad, altamente personalizados (personalización en masa) a un menor costo además de una reducción en el tiempo de mercado y reducción de los ciclos de vida de los productos ya que son rápidamente desechados por una sociedad cada vez más cambiante [1, 19]. Las consecuencias de este escenario para las empresas son las siguientes [20]:

- Aumento de complejidad de productos, dado que aumentan sus características técnicas y sus variaciones.
- Rápido cambio de productos, dada la reducción del ciclo de vida del mismo.
- Rápida introducción del producto, ya que el tiempo de mercado es menor.
- Minimizar la inversión por producto.
- Operaciones globales (geográficamente distribuidas)

Factor Operacional: Los sistemas de control convencional basados en CIM, dedican tiempo para realizar el monitoreo dentro de un sistema jerárquico (de abajo hacia arriba), y luego decidir y aplicar la decisión de control (de arriba hacia abajo) lo cual genera retrasos e inestabilidades [17]. Lo anterior conlleva a una optimización global del sistema pero no responde con eficiencia y eficacia ante cambios dinámicos en los procesos producto de la rigidez de su arquitectura [4]. Tales cambios dinámicos pueden ser: falla de máquinas, ausentismo de personal, cambios de pedidos urgentes, comportamiento impredecible de los clientes y retrasos de proveedores, entre otros [1]. Por otro lado, A pesar de los objetivos y ventajas descritas, la aplicación del concepto de CIM no ha logrado buenos resultados, debido principalmente a problemas tecnológicos (complejidad de automatizar algunos procesos, de heterogeneidad de los sistemas (uso de protocolos propietarios de algunos equipos), sociales (reemplazo de trabajadores por máquinas), económicos (tiene un alto costo de inversión) y de implementación (el enfoque centralizado dificulta la expansión y reconfiguración de un proceso [21].

Dada las condiciones antes mencionadas, se ven justificados grandes esfuerzos por parte de las empresas y de la comunidad científica para optimizar métodos y tecnologías asociadas a la industria de la manufactura con el fin de mejorar la productividad y competitividad [1]. Dentro de este contexto, una gran cantidad de estudios tales como [1, 2, 3, 22, 20] concluyen que las propiedades con las que deben contar los nuevos sistemas de manufactura son:

1. Disponibilidad: capacidad de proveer alta confiabilidad y mantenibilidad independiente del tamaño del sistema y su complejidad.
2. Robustez: capacidad de mantener la operatividad del sistema en caso de un mal funcionamiento.
3. Flexibilidad: esta capacidad es definida en cuatro dimensiones:
  - a. Flexibilidad de mezcla - manejar un rango de productos ejecutando setup en el proceso.
  - b. Flexibilidad de producto – capacidad de modificar rápidamente el diseño del producto.
  - c. Flexibilidad de volumen – cumplir con una producción de volumen variable para satisfacer la demanda.
  - d. Flexibilidad de secuencia – soportar secuencias alternativas para la producción usando recursos que tienen la capacidad de ejecutar diferentes operaciones.
4. Expansibilidad (Escalabilidad): capacidad de añadir o sustraer recursos y/o productos.
5. Reutilización: capacidad para utilizar los equipos de fabricación actuales en la producción de los futuros productos.
6. Reconfiguración: capacidad de soportar distintas estructuras según las condiciones del entorno.

En resumen, las propiedades anteriormente expuestas apuntan a mejorar la confiabilidad y la respuesta dinámica de los sistemas de manufactura ante entornos cambiantes internos y externos.

## 2.5 Sistemas de Control de Manufactura

Los sistemas de control en un sistema productivo pueden ser categorizados en bajo nivel y alto nivel [21]:

**Bajo Nivel:** Se refiere a los dispositivos de automatización como Robots y máquinas de control numérico (CNC). A este nivel las técnicas de control en general son el algoritmo PID (Proporcional-Integral-Derivativo) para variables analógicas y controles On-Off para variables discretas. Ambas estrategias de control, residen en PLC's (Controladores Lógicos Programables) que se encuentran distribuidos o centralizados en la fábrica.

**Alto Nivel:** Se refiere a la coordinación de recursos para producir un producto. A este nivel los algoritmos resuelven los problemas asociados a qué producir, cuanto producir, cuando producir, cuando la producción será finalizada, cómo y cuándo se usarán los recursos o hacerlos disponibles, cuando se libera un trabajo dentro de la fábrica, cuales trabajos serán liberados, la ruta del trabajo y la secuencia del trabajo/operación. Es en este tipo de control en donde la presente tesis coloca su énfasis.

Dada las propiedades exigidas a los nuevos sistemas de manufactura indicadas en el punto 2.4, la comunidad científica ha llegado a la conclusión que los nuevos paradigmas asociadas a los sistemas de control de manufactura deben exhibir las siguientes características [20, 23, 11, 1]:

- ✓ La arquitectura de control debe ser descentralizada y basada en el producto/recurso.
- ✓ Las decisiones de control global son determinadas por más de una unidad de control.
- ✓ Las interacciones de control debe ser abstractas, generalizadas y flexibles.
- ✓ El control debe ser reactivo y pro-activo.
- ✓ El control debe ser auto-organizado.

**La arquitectura de control debe ser descentralizada y basada en el producto/recurso -**

Un problema complejo se puede dividir en varios problemas pequeños, cada uno asignado en una unidad de control. Cada unidad de control es autónoma con sus propios objetivos, conocimientos y habilidades, y encapsula funciones inteligentes; sin embargo, ninguno de ellos tiene una visión global del sistema. Con el fin de permitir que la arquitectura sea flexible, de complejidad reducida y además facilite la reconfiguración del sistema, la introducción y la eliminación de productos y/o recursos (escalabilidad) la arquitectura debe ser basada en productos y recursos. Cada recurso debe contener todas las funciones de control necesarias para procesar los trabajos mientras que cada orden y pieza de trabajo debe tener las capacidades de control para ser producida.

**Las decisiones de control global son determinadas por más de una unidad de control –**

Las tareas como, por ejemplo, la programación, el seguimiento y diagnóstico son tomadas por el conjunto de entidades decisionales (Ej.; ERP, MRP, MES, DCS, PLC), interactuando en un entorno de colaboración para llegar a una decisión de producción.

**Las interacciones de control debe ser abstractas, generalizadas y flexibles -** Para facilitar la escalabilidad y la re-configurabilidad, los recursos deben desacoplarse en tres pasos:

Abstracción de la interacción - un componente no necesita del conocimiento del funcionamiento interno de otro componente.

Interacción generalizada – un componente no necesita un conocimiento preciso del comportamiento de otro componente.

Relaciones e interacciones flexibles - cada componente decide dinámicamente con quien interactuar.

**El control debe ser reactivo y pro-activo** - Con el fin de responder a los cambios y perturbaciones a corto plazo, el control debe ser reactivo. Además, el sistema debe ser pro-activo dado que debe contar con capacidades de planificación distribuida y debe anticiparse, evaluar situaciones y prepararse ante situaciones críticas.

**El control debe ser auto-organizado** – El sistema debe responder ante cualquier perturbación aunque esta implique una re-configuración de recursos y/o el reordenamiento del proceso en forma automática.

En la Figura 8 se detalla un resumen de las propiedades con las que debe contar un sistema de manufactura y su implicancia en los requerimientos impuestos al sistema de control.

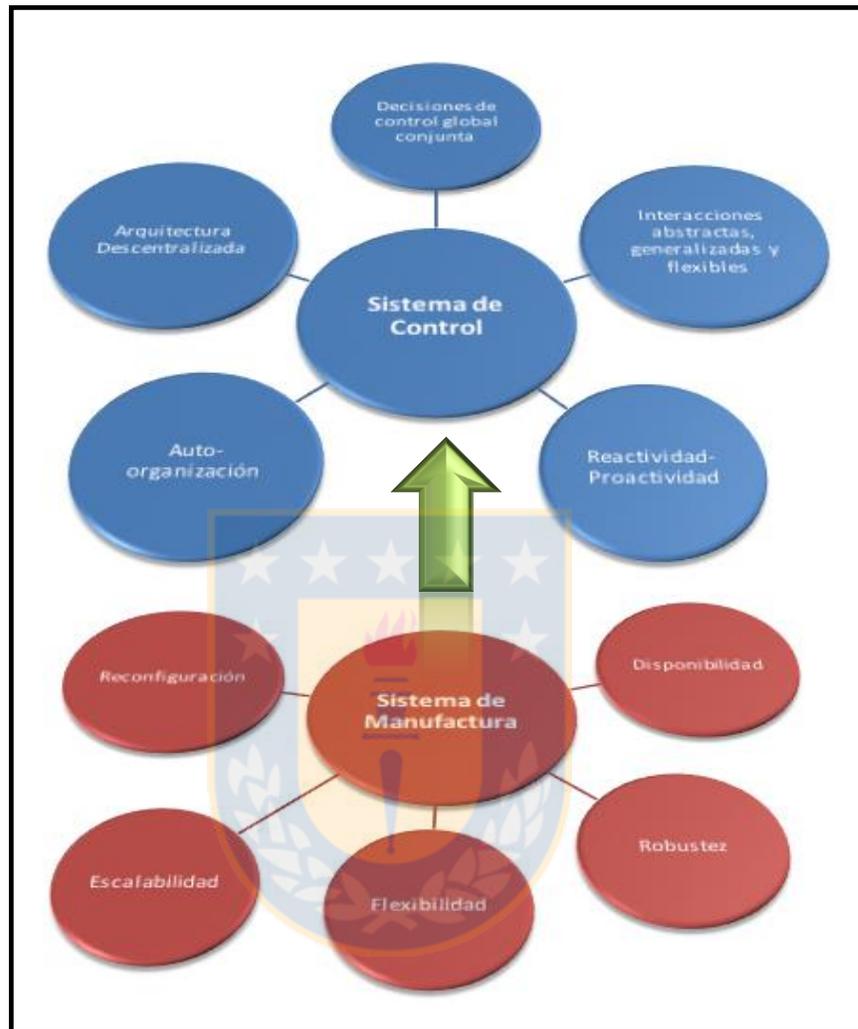


Figura 8: Resumen de Propiedades a exhibir por el sistema de manufactura y requisitos para su sistema de control

## 2.6 Enfoques y paradigmas emergentes para el Sistemas de Control de Manufactura

Dada las características que deben exhibir los nuevos sistemas de control expuestas en el punto 2.5, desde los años 90', se comienzan a desarrollar nuevos enfoques basados en la distribución de la toma de control. La idea era permitir que las entidades de toma de decisiones trabajaran juntas con el fin de reaccionar rápidamente en lugar de solicitar decisiones de control de los niveles de toma de decisiones superiores, las cuales generan tiempos de respuesta tardíos. En este nuevo enfoque de la distribución de control, aparecen distintos procesos de interacción distinta tales como la coordinación, la negociación y la cooperación.

Las diferentes formas de distribuir las decisiones de control desde los sistemas de control centralizados hasta sistemas de control no centralizados se basan en dos opciones de diseño: relaciones jerárquicas y relaciones heterárquicas. En la gráfica mostrada en la Figura 9 se observan las distintas arquitecturas posibles según la clasificación utilizada por [17] y ratificada por [1], las cuales se describen a continuación:

**Clase I - Arquitectura Totalmente Jerárquica:** se caracteriza por la existencia de varios niveles de control, lo que permite la distribución de la toma de decisiones entre los niveles jerárquicos, la introducción de una mejor robustez, la previsibilidad y la eficiencia. Sin embargo, la aparición de perturbaciones en el sistema reduce significativamente su rendimiento (Arquitectura utilizada en el paradigma CIM).

**Clase II - Arquitectura Jerárquica Modificada:** intenta mejorar la respuesta a las perturbaciones, manteniendo todas las características de la arquitectura jerárquica y la adición de la interacción entre los módulos en el mismo nivel jerárquico. La expansibilidad del sistema es más fácil que en la arquitectura jerárquica debido a la característica de interacción en el mismo del nivel de control.

**Clase III - Arquitectura Totalmente Heterárquica:** alto rendimiento frente a las perturbaciones, la optimización global se verá reducida mientras que la optimización local se ve beneficiada, esto debido a que la toma de decisiones es local y autónoma, sin visión del sistema global. La expansibilidad del sistema es una tarea más fácil, porque es suficiente modificar sólo el funcionamiento de algunos módulos o añadir nuevos módulos al sistema de control.

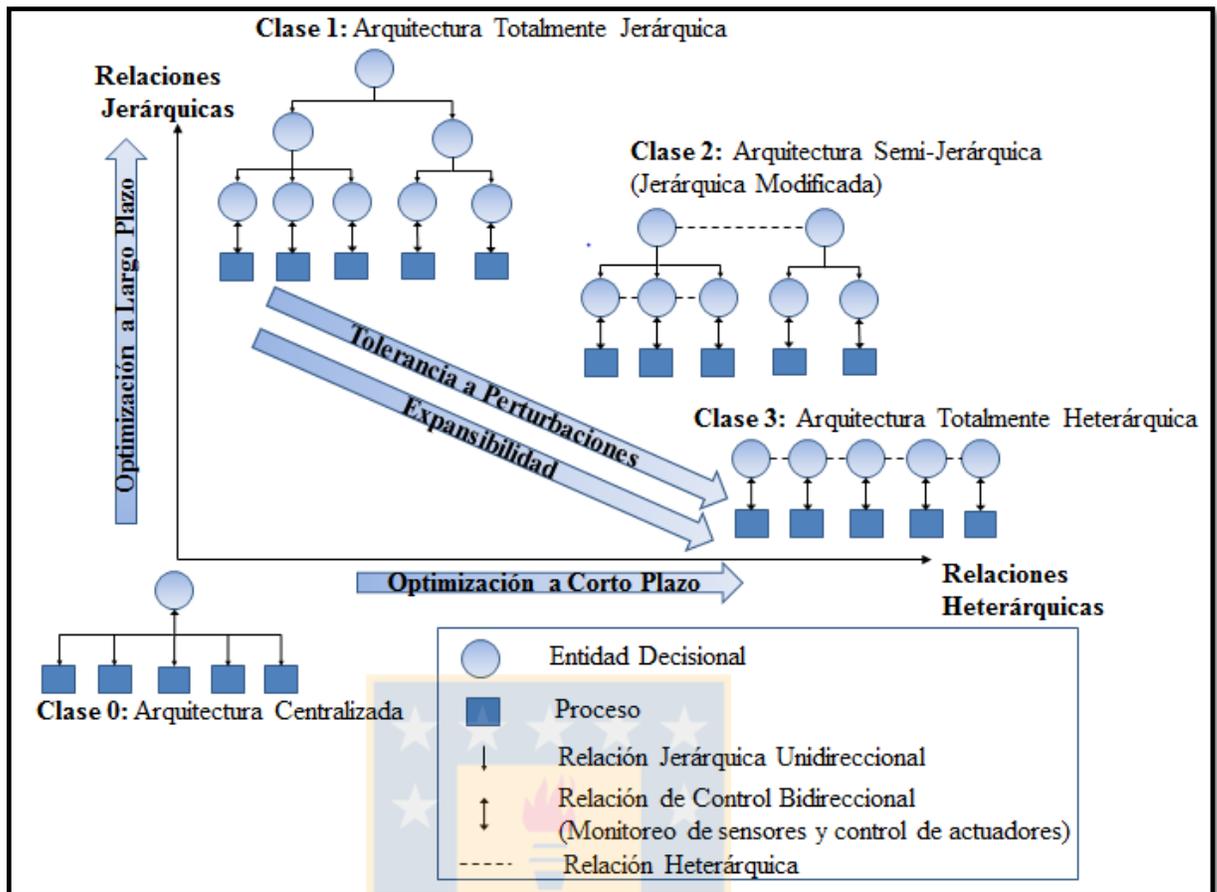


Figura 9: Clasificación de arquitecturas de control y sus implicancias. Imagen adaptada de [17]

En la Tabla 1, se resume un cuadro comparativo que muestra las diferencias entre un sistema de control distribuido respecto de un control convencional.

**Tabla 1: Tabla comparativa control convencional vs Control Distribuido [1]**

<b>Control Convencional</b>	<b>Control Distribuido</b>
Solución centralizada y jerárquica.	Solución distribuida con relaciones jerárquicas y/o heterárquicas con cooperación
Arquitectura rígida y estática.	Arquitectura flexible y dinámica
Relaciones Cliente-Servidor.	Relaciones Entidad Decisional- Entidad Decisional
Tiempo para encontrar la solución restringido por la potencia de procesamiento.	Tiempo para encontrar la solución restringido por la velocidad de comunicación.
Enfoque de arriba hacia abajo.	Enfoque de abajo hacia arriba
Eficiencia a través de la especialización.	Eficiencia a través de la flexibilización.
Débil respuesta a perturbaciones.	Alta respuesta a las perturbaciones.
Operadores son reemplazados por la automatización (removido del proceso)	Los operadores son complementados con la automatización (operadores incrementan sus habilidades y se mantienen en el proceso).
Más eficiente para alto volumen de producción y baja variabilidad de productos	Más adecuado para alto y bajo volumen de producción y alta variabilidad de productos.

Uno de los enfoques que ha cobrado fuerza los últimos años es la arquitectura heterárquica basada en productos inteligentes (Product Driven Control Systems, PDCS). La idea detrás de este concepto es que el centro de la solución de la automatización está en el flujo del producto, lo cual es opuesto al enfoque convencional orientado al recurso para automatizar células de manufactura [8]. La inteligencia y la autonomía se traen lo más cerca posible del sistema físico, o incluso embebida en el mismo producto a ser manufacturado que, en general, no consideran ninguna inteligencia (productos pasivos). Lo anterior permite a las

entidades de decisión trabajar y colaborar juntas con el objetivo de reaccionar rápida y autónomamente (con ciertas restricciones), en vez de solicitar decisiones de control a un nivel más alto de decisión el cual genera retardos de tiempos de respuesta. Algunos trabajos desarrollados con este enfoque pueden ser revisados en [8], [5], [9] y [10].

Otro enfoque ampliamente estudiado son los sistemas de arquitectura híbrida. Los estudios que desarrollan este tipo de enfoques [9] indican que si bien las arquitecturas totalmente heterárquicas evidencian ser sistemas altamente reactivas ante decisiones locales, a la vez, evidencian que cada entidad decisional presenta dificultad para predecir su propio comportamiento en el futuro y el de las demás entidades, conllevando a la obtención de soluciones no óptimas a mediano y largo plazo, este síntoma es denominado Miopía. Con el fin de mitigar el efecto de la miopía, se desarrollan sistemas de control híbridos, es decir, desarrollar sistemas que se comporten de forma centralizada cuando el sistema está estable y de forma descentralizada cuando el sistema es perturbado, ejemplos de lo anterior son ADACOR [11], ADACOR Evolution [24] y el trabajo desarrollado en [9]. Los resultados obtenidos en la aplicación de estos sistemas, demuestra que las arquitecturas híbridas son más eficientes que las arquitecturas estáticas ya que aprovechan las ventajas de optimización global de una arquitectura jerárquica y la reactividad de una arquitectura heterárquica cuando existe una perturbación en el sistema [11].

Varios paradigmas se han introducido al contexto de los sistemas de manufactura en los últimos años, los cuales se adaptan a los nuevos enfoques de arquitecturas y a las propiedades impuestas para los sistemas de control, tales como:

- Modelo de Sistemas Viables (VSM) [5].
- Sistemas Multi-agentes (MAS) [6].
- Sistemas Holónicos de Manufactura (HMS) [7].

En los siguientes puntos se detallan los paradigmas anteriormente mencionados.

### 2.6.1 Modelo de Sistemas Viables (VSM)

El modelo de sistemas viables propuesto por Stanfford Beer en 1979 [25] supone que cada sistema está relacionado con otros sistemas ubicados a niveles de observación más elevados. Toda entidad puede ser entendida como un sistema y puede ser considerada un micro-entorno formado por un conjunto de sub-componentes interrelacionados orientados a un fin común, condición necesaria para considerar al elemento agregado como un sistema.

Herrera en su trabajo [5] presenta un marco de referencia para los sistemas de control de manufactura basados en el producto, aplicando el paradigma de sistemas viables. Comenta que la propiedad principal en VSM es la recursividad, lo que permite gestionar la complejidad y encontrar un acuerdo entre el control y la autonomía para aplicar los principios como heterarquía y la cooperación, siendo estos los elementos centrales de los sistemas de control basados en productos y términos más generales la clave de los IMS.

En la Figura 10, se muestra el diseño del sistema de control el cual se descompone en cuatro niveles recursivos de entidades de productos. En cada nivel, las entidades se modelan como un producto inteligente. Para tomar la decisión, un producto o un grupo de productos tendrán objetivos, restricciones, inteligencia y funciones de control además de mecanismos de coordinación y comunicación. Como conclusión del trabajo se comenta que la principal ventaja del modelo aplicado es alcanzar una coherencia entre los diferentes niveles de decisión dando un alto grado de flexibilidad al sistema.

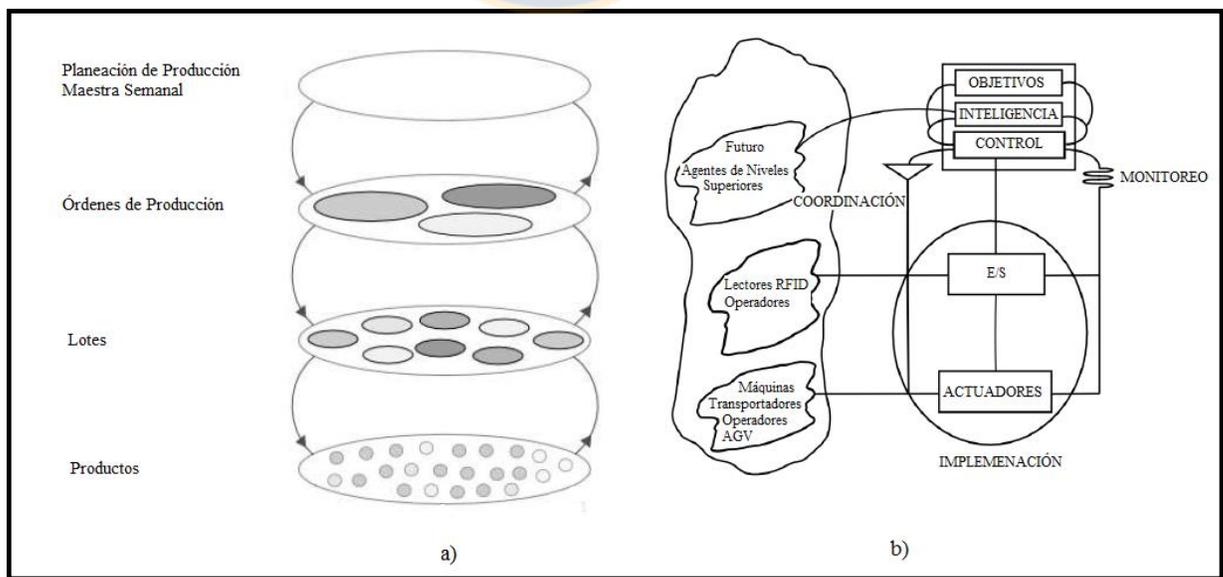


Figura 10: a) Diseño de Sistema de Control basado en VSM. b) Estructura de un producto Inteligente usando VSM  
Imagen adaptada de [5]

## 2.6.2 Sistemas Multi-agentes (MAS) y Sistemas holónicos de Manufactura (HMS)

El paradigma de sistema multi-agente (Multi Agent System, MAS) se deriva del campo de la inteligencia artificial distribuida (DAI). Un sistema multi-agente puede ser definido como un conjunto de agentes que representan los objetos de un sistema, capaces de interactuar, con el fin de alcanzar sus metas individuales, cuando no tienen suficientes conocimientos y/o habilidades para lograr sus objetivos de forma individual. Las características principales de un agente son la adaptación, cooperación y la coordinación con el fin de lograr un objetivo. Es por este motivo que los sistemas multi-agente han cobrado gran relevancia en el desarrollo de las nuevas arquitecturas de sist. de control de manufactura [1].

El paradigma MAS reemplaza el control centralizado por una funcionalidad distribuida donde las interacciones entre las entidades, conducen a la aparición de un comportamiento global inteligente. Esto permite alcanzar un alto grado de autonomía y cooperación sin una estructura cliente-servidor [6].

Estos sistemas multi-agentes han sido ampliamente utilizados para modelar sistemas de control tanto Jerárquico Modificado como totalmente heterárquicos. Sin embargo, desde finales de 1990, un nuevo paradigma surgió: el paradigma holónico.

El paradigma holónico está basado en el concepto desarrollado en 1969 por Koestler [26] y luego fue propuesto en el contexto de la industria de la manufactura en 1994 por el consorcio HMS como caso de estudio de caso bajo el programa de investigación internacional para los Sistemas Inteligentes de Manufactura (“Intelligent Manufacturing Systems”, IMS) [7]. El deseo de integrar mecanismos tanto jerárquicas y heterárquicas en un sistema de control distribuido puede ser visto como una característica esencial del paradigma holónico, permitiendo a los sistemas beneficiarse de las ventajas de ambos enfoques.

Las principales características de un holón son [1] [27]:

- **Autonomía**, donde la estabilidad de los holones es el resultado de su capacidad para actuar de forma autónoma en caso de circunstancias impredecibles.
- **Reactividad**, los holones son capaces de percibir estímulos de su entorno y reaccionar ante ellos.
- **Proactividad**, los holones pueden actuar guiados por sus propios objetivos.

- **Cooperación**, capacidad de tener holones cooperando, transformando estos holones en componentes eficaces de un sistema mayor.
- **Representación Física/Lógica**, un holón puede representar una actividad física o lógica, como por ejemplo un robot, una máquina, una orden, un sistema de fabricación flexible o incluso un operador humano. El Holón tiene información sobre sí mismo y el medio ambiente, el cual contiene una parte de procesamiento de información y una parte de procesamiento físico solo cuando el holón representa un dispositivo físico.
- **Holarquía**, se define como un sistema de holones organizados en una estructura jerárquica, cooperando para alcanzar los objetivos del sistema, mediante la combinación de sus habilidades y conocimientos individuales. Cada holarquía ha fijado normas y directivas. Un holón puede pertenecer a múltiples holarquías dinámicamente al mismo tiempo, lo cual es una diferencia importante con el concepto tradicional de las jerarquías. Los holones pueden integrarse en un holarquía y, al mismo tiempo, preservar su autonomía e individualidad.
- **Arquitectura Abierta**, el sistema está habilitado para crear o eliminar holones, modificar sus capacidades funcionales.

La primera arquitectura general de un holón fue propuesta por Christensen [7] y luego ratificada en [28] como ampliamente aceptada por la comunidad. En este modelo se define al holón de acuerdo a la siguiente estructura de bloques (ver Figura 11):

- Interfaz holón-holón y humana, permite al holón interactuar con el exterior.
- Toma de decisiones, corresponde al conocimiento y el status del holón para tomar decisiones.
- Control Físico, controla al elemento físico.
- Elemento Físico, representa la ejecución de operaciones de manufactura.

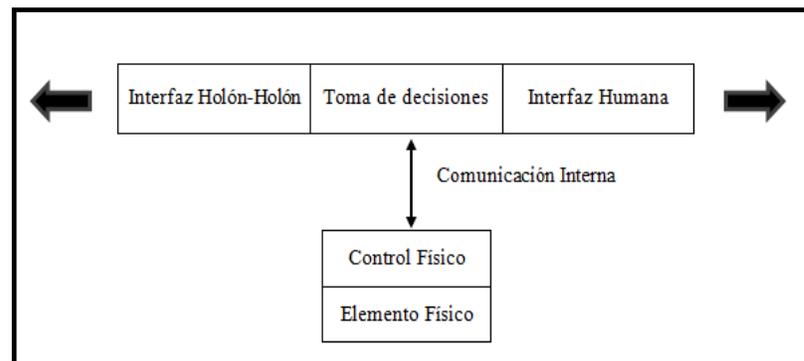


Figura 11: Arquitectura de un Holón. Imagen adaptada de [28].

### 2.6.2.1 Diferencias entre MAS y HMS

Gran cantidad de publicaciones científicas han debatido respecto a las similitudes y diferencias entre un MAS y un HMS [7] , [26], y [29]. En [1], se describen estas diferencias en términos de origen, conceptual y de modelamiento. En términos de origen los agentes nacen de las ciencias de la computación, mientras que los holones tienen su origen en los sistemas sociales y marco de trabajo fue impulsado para ser aplicado específicamente para cumplir con los nuevos requerimientos de los sistemas de manufactura. En términos conceptuales un agente es un paradigma de programación y una tecnología mientras que el holón es un concepto, el cual es posible implementar a través de la tecnología de agentes (es el caso de las plataformas para desarrollo de agentes como JADE). En términos de Modelamiento, los agentes en general, representan un componente de software sin enfocarse en la integración de dispositivos físicos, mientras que el concepto de holón soporta la integración de dispositivos físicos. Una comparación más en detalle es posible analizar en [27], a continuación se resumen las comparaciones más relevantes tratadas en este trabajo entre MAS y HMS.

<b>Propiedad</b>	<b>Holón</b>	<b>Agente</b>
<b>Autonomía</b>	Sí	Sí
<b>Reactividad</b>	Sí	Sí
<b>Proactividad</b>	Sí	Sí
<b>Habilidad Social</b>	Sí, cada holón cuenta con su interfaz humana	Sí, la interfaz humana, generalmente es implementada por uno o más agentes especializados
<b>Cooperación</b>	Sí, un holón nunca rechazada cooperar.	Solo cuando le conviene (desde el punto de vista de sus objetivos).
<b>Mobilidad</b>	Sí, pero rara vez la necesitan.	Sí
<b>Recursividad</b>	Sí	No, pero se pueden usar algunas técnicas para que cuente con esta propiedad (mediadores, facilitadores, etc).
<b>Representación Físico/Lógico</b>	Sí, hay una separación explícita, la parte física es opcional.	No existe una separación explícita.

Las conclusiones comentadas en los trabajos y [29], tienen como denominador común que un holón puede ser visto como un caso particular de agente. Mientras que MAS es una concepto y tecnología aplicable a varios campos, los HMS son un concepto construido especialmente para implementar sistemas de manufactura inteligentes y que puede utilizar la tecnología MAS disponible para ser implementado.



### 2.6.2.2 Principales arquitecturas con enfoque holónico

A continuación se describen las principales arquitecturas de sistemas holónicos con mayor atención en la comunidad científica:

**PROSA (Product-Resource-Orden-Staff-Architecture)** [30]. Esta es una de las primeras arquitecturas de referencia genérica para la capa de control que cubre aspectos de los enfoques jerárquicos y heterárquicos y, al mismo tiempo, reduce la complejidad de integrar nuevos componentes, facilitando la reconfiguración del sistema. Está compuesta de tres holones básicos los cuales usan los conceptos de agregación y especialización (Ver Figura 12).

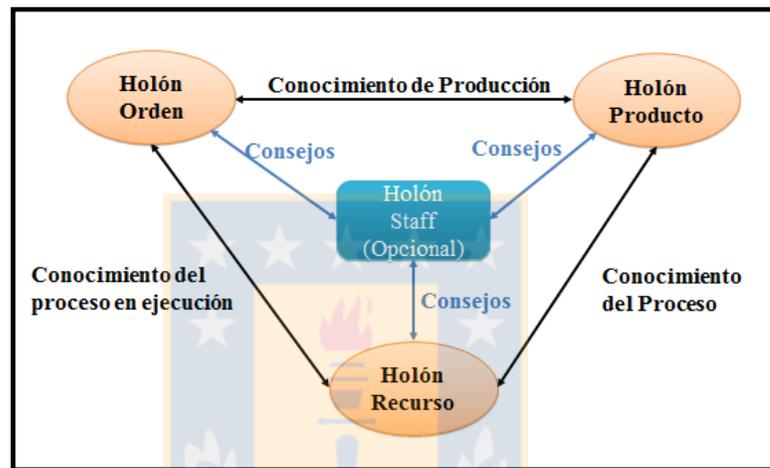


Figura 12: Arquitectura de Referencia PROSA  
Fuente: Imagen adaptada de [30].

A continuación se describen las entidades que son parte de su arquitectura:

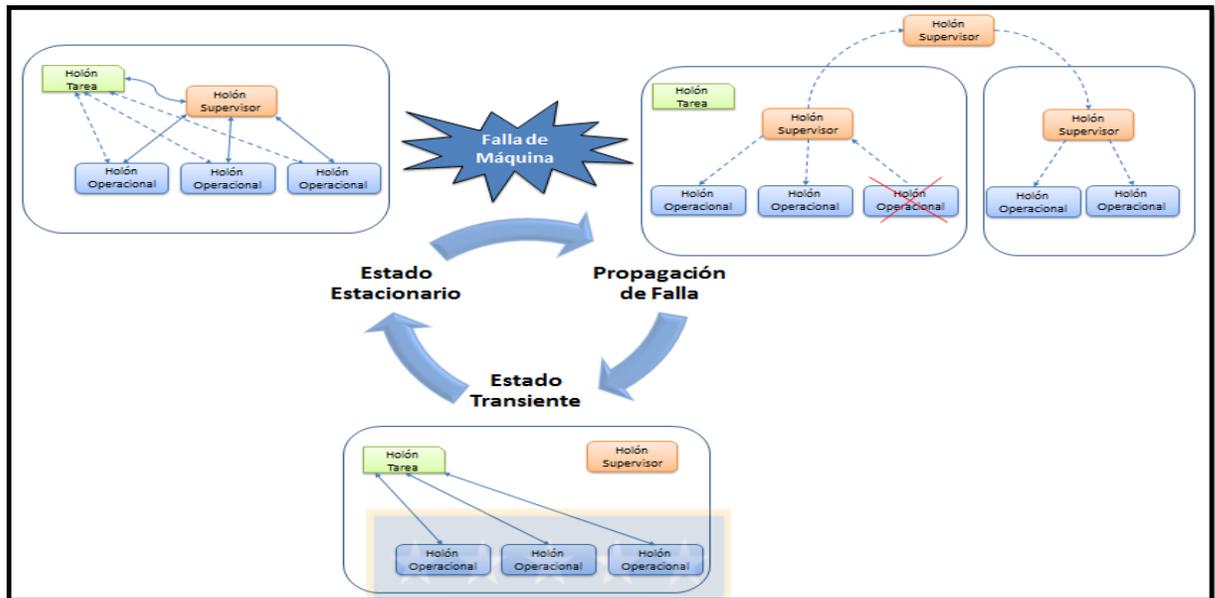
**Holón Producto** : mantiene el modelo del producto y el conocimiento acerca del plan de proceso para su producción. Este holón es un servidor de datos para otros holones.

**Holón Recurso** : mantiene los métodos para asignar los recursos a otros holones, el conocimiento y los procedimientos para organizarlos.

**Holón Orden** : representa la tarea para hacer el producto de acuerdo a los requerimientos del cliente y el responsable de que el trabajo asignado se realice correctamente y a tiempo. Maneja los estados del producto físico, controles logísticos y operaciones de producción dada la negociación con los holones recurso.

**Holón Staff** : es una entidad opcional, considerado un experto externo que da consejos a los holones básicos con respecto a la solución correcta a un problema.

**ADACOR (Adaptive Holonic Control Architecture)** [23, 11]: este enfoque introduce la colaboración y la reconfiguración automática para un sistema de manufactura distribuido desde un estado estacionario a un estado transiente (ver Figura 13).



**Figura 13: Reorganización Dinámica de ADACOR. Imagen adaptada de [11]**

El foco de esta arquitectura es el nivel de piso de la planta y específicamente los sistemas de manufactura flexibles organizados en un sistema de producción tipo Job Shop, basado en el paradigma holónico y específicamente en la arquitectura de referencia PROSA, ADACOR realiza un enfoque centralizado cuando el objetivo es la optimización y un enfoque descentralizado cuando existe un evento inesperado. Actualmente, se propone la evolución de ADACOR denominado ADACOR<sup>2</sup> [24], la diferencia con la primera versión de ADACOR radica en que se tienen dos tipos de reorganización: suave y radical. La reorganización suave es lograda a través de un estabilizador de nervios que es implementado a través de un algoritmo PID (Proporcional- Integral-Derivativo). A continuación se describen sus holones básicos:

Holón Producto : ídem al Holón Producto definido en la arquitectura PROSA.

Holón Tarea : ídem al Holón Orden definido en la arquitectura PROSA.

Holón Operacional : ídem al Holón Recurso definido en la arquitectura PROSA.

Holón Supervisor : es un coordinador que propone planes de programación optimizados para los holones tarea y operacional. Este holón está habilitado para propagar la necesidad de

reorganización a los holones tarea para encontrar planes alternativos de programación si se detecta una perturbación.

Una reciente aplicación de las arquitecturas de referencia PROSA y ADACOR se propone en [8] para implementar una arquitectura de control basada en el producto. Los holones órdenes son equivalentes a los productos inteligentes los cuales toman decisiones acerca de su ruta de manufactura en tiempo real y en forma automática. En este trabajo se proponen dos estrategias heterárquicas: La primera basada en productos con inteligencia local y la segunda basada en productos con inteligencia remota distribuida en la red de computadores. De este trabajo se concluye que un control descentralizado incrementa la reactividad y provee robustez ante perturbaciones, por otro lado, se comenta además que un sistema basado en productos con inteligencia remota es más eficiente que un sistema basado en productos con inteligencia local dentro de un ambiente altamente colaborativo, por otro lado, se indica que los productos con inteligencia local pueden ser una alternativa en casos de tareas simples.

**HCBA (Arquitectura basada en componentes holónicos) [31]:** se deriva de la CDB (desarrollo basado en componentes) y HMS. Se proponen bloques inteligentes de recursos y de productos para formar dinámicamente un controlador virtual a través de una red informática, llevando a cabo operaciones cooperativas de ejecución de control y diagnóstico. Este concepto permite el diseño, funcionamiento y mantenimiento del controlador de manufactura de forma distribuida, lo cual puede aumentar la agilidad y la capacidad de respuesta de un sistema integrado. A continuación se describen sus holones básicos:

**Holón Recurso** : es un componente del sistema integrado que puede ejecutar operaciones tales como la producción, montaje y transporte.

**Holón Producto** : puede contener una parte física y una parte de control. La parte física puede ser, por ejemplo, materias primas o las piezas de un producto. La parte de control puede representar el control de la ruta de una línea de producción, control de procesos, toma de decisiones y/o la información del producto.

## 2.7 Aplicaciones de Sistemas Multi-agentes y Holónicos

Una gran cantidad de arquitecturas y aplicaciones asociadas a sistemas MAS y HMS han sido desarrolladas y se resumen en la Figura 14.



Figura 14: Evolución de arquitecturas y aplicaciones basadas en agentes y su aplicación en la industria.  
Fuente: Imagen extraída de [6]

A continuación se comentan algunas aplicaciones reales desarrolladas por sistemas MAS y HMS.

**DaimlerChrysler:** quien desarrolló el proyecto Production 2000+ el cual consistió en un sistema de producción flexible de culatas que fue instalado en una fábrica real como bypass en una parte de la línea de producción existente.

**Rockwell Automation :** ha encabezado algunos proyectos tales como un sistema de agentes que ayudó a aumentar la utilización de una máquina en BHP Billiton en Australia y un sistema automático de enfriamiento de agua de un buque de la marina de Estados Unidos [12]. Además desarrolló una plataforma de simulación basada en agentes denominada MAST [32], e incluso fue utilizada para realizar control físico en una célula de trabajo en el centro para la automatización y control distribuido de la Universidad de Cambridge.

**NASA:** usó tecnología basada en agentes para balancear las múltiples demandas de satélites, como mantenerse en curso y hacer frente a situaciones inesperadas [6].

A continuación en la Tabla 2 se detallan distintas arquitecturas desarrolladas y sus ámbitos de aplicación en el contexto de un sistema de manufactura. Una lista detallada de aplicaciones puede ser revisada en [33]:

**Tabla 2: Aplicación de sistemas MAS y HMS según ámbito de aplicación**

Arquitectura	Referencia	Ámbito de Aplicación			
		Colaboración Empresarial	Integración Empresarial	Planificación y Programación	Control de Manufactura
PROSA	[30]				X
ADACOR	[11]			X	X
HCBA	[31]				X
EXPLANTECH	[34]			X	
METAMORPH I	[35]		X	X	
METAMORPH II	[36]	X	X	X	
MASCADA	[37]			X	X
VSM-SCP	[5]			X	X
Sin nombre	[8]				X

## 2.8 Simulación basada en Agentes

La experimentación de sistemas de control holónicos en sistemas físicos pudiese ser costosa y en las condiciones del hardware actual prácticamente irreal, por lo tanto, se torna fundamental el hecho de contar con plataformas de simulación. Esta plataforma debe contar de dos partes:

1. Herramienta de simulación para correr el modelo de un proceso: existe una multitud de software para programar el modelo de un proceso tanto continuo como discreto, ejemplos de estos son: Matlab, Arena Rockwell y LabView.
2. Un Entorno de ejecución de agentes: aplicación desarrollada para ejecutar a los agentes y modelar sus acciones tales como: JADE, SPADE, ZEUS. Este software es capaz de proveer un marco de trabajo que asegure: la administración de Agentes, envío de mensajes y comunicación, movilidad, robustez y en general un ambiente que funcione como esqueleto para un sistema multi-agente. Según un estudio de aplicaciones multi-agentes [38], considerando las características más relevante de una plataforma para el desarrollo de sistemas multi-gerentes tales como: naturaleza del producto (libre o comercial), estándar implementado, técnicas de comunicación, mecanismos de seguridad, movilidad de agentes y mecanismos de migración (Ej. RMI o Socket), la plataforma que mejor performance cumple es JADE.

En la Tabla 3 se puede observar algunas características relevantes de los aplicaciones libres más utilizados por la comunidad científica. También se encuentran algunas aplicaciones comerciales como Cybele y Voyager.

Tabla 3: Plataformas de Ejecución de Sistemas basados en Agentes

Plataforma	Licencia	Lenguaje de Programación	Estándar	Comunicación entre agentes
JADE	Libre	Java	FIPA	ACL*
SPADE	Libre	Python	FIPA XMPP/Jabber	ACL
FIPA-OS	Libre	Java	FIPA	ACL
Zeus	Libre	Java	FIPA	ACL/KQML**

\*: Agent Communication Language; \*\*: Knowledge Query and Manipulation Language.

### 2.8.1.1 Direcciones de Investigación Futura en el campo de los sistemas Inteligentes de Manufactura

Múltiples tareas son necesarias para que los nuevos paradigmas sean implementados en la industria de forma masiva, en [6] se resumen algunos aspectos importantes a considerar por la comunidad científica para implementar estos sistemas:

1. **Proveer demostraciones corriendo en la industria.** Hay que permitir que las empresas comiencen a creer en la nueva tecnología. Ya existen proveedores de sistemas de automatización que están invirtiendo en el desarrollo de sistemas de manufactura inteligente tales como Rockwell Automation, Siemens y Schneider Electric e incluso asociadas con empresas desarrolladoras de sistemas ERP como SAP. Ejemplos de trabajo en conjunto son los proyectos como SOCRATES e IMC-AESOP.
2. **Proveer un análisis del retorno de la inversión (ROI).** Este análisis debe considerar los costos asociados a desarrollo, operación y mantención de soluciones basadas en agentes.
3. **Proveer soluciones basadas en agentes como caja negra.** La complejidad de los sistemas basados en agentes debe ser transparente para el usuario, por tanto, es necesario desarrollar interfaces estándares y de fácil configuración.
4. **Desarrollo de metodología y tecnologías** Desarrollo de sistemas basados en conocimiento interoperable. Actualmente se trabaja en un estándar de bloques de función IEC 61499 como una evolución del actual estándar IEC 61131 para programación de PLC. El estándar IEC 61499 es un enfoque para modelar el control distribuido y habilitar a las soluciones basadas en agente ser útiles en aplicaciones de tiempo real. Sin embargo, este estándar no es soportado por los proveedores de PLC, ni existen herramientas de configuración. Un desarrollo importante en este sentido es el trabajo desarrollado por Rockwell Automation el cual desarrollo un firmware que habilita utilizar el paradigma holónicos en su gama más alta de PLC: ControlLogix [12] [32].
5. **Herramientas de Simulación estándares.** Actualmente la herramienta más utilizada para desarrollar sistemas Holónicos es JADE, sin embargo, no existen metodologías estándares para el desarrollo de sistemas y simulación de procesos. La utilización de herramientas de simulación es una poderosa forma de probar aplicaciones basadas en agentes, permitiendo evaluar la performance de los sistemas propuestos.

## 2.9 Brecha detectada

Si bien existen múltiples aplicaciones de los sistemas holónicos tanto en ambientes físicos como simulados, no existen trabajos científicos que apliquen el paradigma holónico al control de procesos simulados a través de redes de cola o que embarquen el modelo de redes de cola como motor de inteligencia para un sistema de control holónico real de un proceso.

Las Redes de Cola son utilizadas para simular diversos procesos discretos desde redes de comunicaciones, tráfico vehicular, tráfico aéreo, sistemas hospitalarios hasta procesos de manufactura como la industria automotriz, salmoneras, producción de tableros, producción de Sistemas Informáticos (Notebooks, PC, etc.), redes informáticas, etc. [39].

Esta brecha detectada da pie a las siguientes líneas de investigación:

1. Un proceso simulado a través de redes de cola y controlado a través de un sistema de control holónico podría ser una alternativa para probar sistemas de control antes de su implementación.
2. Un sistema de control holónico que embarque la inteligencia del modelo de redes podría reemplazar a un sistema de control convencional de un proceso real dando mayor robustez ante cambios dinámicos y perturbaciones del entorno.
3. Un sistema de control holónico que embarque la inteligencia del modelo de redes de cola de un proceso podría ser aplicado en paralelo a un sistema de control convencional para predecir comportamientos futuro dada una falla en el sistema de manera de tomar acciones anticipativas.

El presente trabajo busca proponer una solución para la línea de investigación N°1 anteriormente definida. Sin embargo, la propuesta puede ser usada como base para desarrollar las líneas de investigación N°2 y N°3.

## CAPÍTULO 3

### Materiales y Métodos

#### 3.1 Introducción al capítulo

En el capítulo anterior se concluye que los sistemas distribuidos e inteligentes y, más en concreto, los sistemas de control holónicos de manufactura presentan ventajas significativas respecto a los sistemas de control de manufactura convencionales. Sin embargo, los sistemas de control holónico no han sido explorados ni aplicados a procesos desde un punto de vista de redes de cola.

En el presente capítulo se plantea una arquitectura de control holónica que integra a su inteligencia el modelo de redes de cola de un proceso en particular. Esta arquitectura es validada a través de experimentación utilizando un entorno en donde los holones sociabilizan y una herramienta de simulación de redes de cola.

#### 3.2 Arquitectura propuesta

Se propone un sistema de control de red de cola con un enfoque Holónico, en adelante CRC-H (**C**ontrol de **R**edes de **C**ola **H**olónico), tomando en consideración los conceptos tales como: control distribuido, cooperación y reorganización entre las entidades de control. La arquitectura del sistema de control es dinámica, es decir, se reorganiza de acuerdo al estado operacional del sistema, idea que ya aplica Leitão en la arquitectura de ADACOR [11]. Se propone una arquitectura centralizada cuando no existe falla en el sistema y por lo tanto, será la arquitectura por defecto en la que opera el proceso, sin embargo, cuando existe una falla, la arquitectura centralizada se convierte en una arquitectura Heterárquica (clase 3) distribuyendo el control y la toma de decisiones. Una vez recuperada la falla el sistema vuelve a una arquitectura centralizada, este proceso se describe gráficamente en la Figura 15. A continuación se detalla en profundidad ambas arquitecturas:

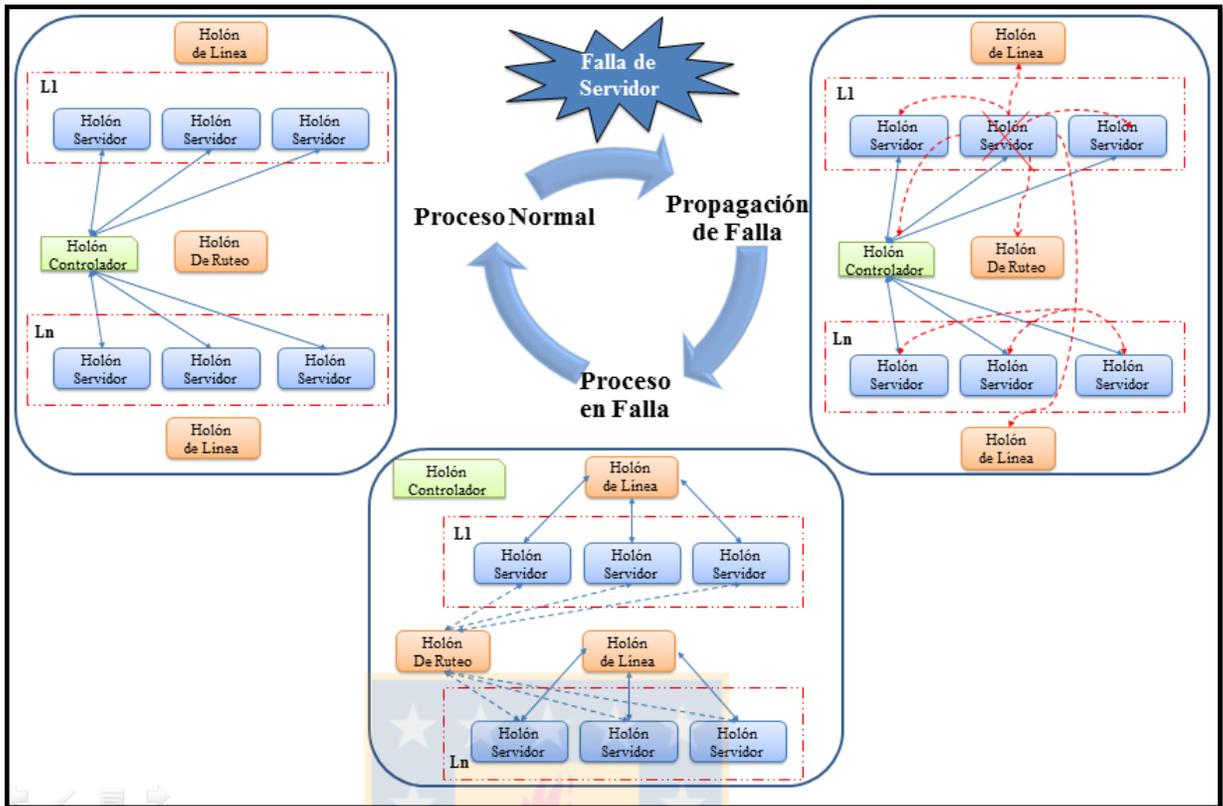


Figura 15: Etapas de transición de arquitecturas.  
Fuente: Elaboración propia.

1. Arquitectura Centralizada: solo existe una entidad decisional, la cual contiene el modelo matemático que optimiza el proceso (Función a Maximizar/Minimizar y Restricciones). Las tasas de servicio de cada servidor son controladas por una única entidad en función del modelo matemático que optimiza la producción de acuerdo métricas preestablecidas.
2. Arquitectura Heterárquica: existe una comunidad de Holones los cuales interactúan entre ellos compartiendo información para tomar decisiones a nivel local. La idea fundamental de esta arquitectura es que dada una falla el sistema realice las siguientes tareas:
  - a. Controlar el ruteo de trabajos de forma inteligente. El ruteo inteligente se realiza modificando las probabilidades de ruta en función de las colas de trabajo en el primer servidor de cada línea en paralelo.
  - b. Controlar las tasas de producción de forma inteligente. El control inteligente de tasas de servicio consiste disminuir las tasas de los servidores de la línea con un

servidor en falla a la tasa de producción de este último. Las tasas de servicio de los servidores de las otras líneas se mantienen sin cambios.

A continuación, en la Tabla 4 se describen las entidades fundamentales que definen la arquitectura heterárquica:

**Tabla 4: Descripción de Holones que componen el sistema de control propuesto**

<b>Entidad</b>	<b>Arquitectura</b>	<b>Función</b>
<b>Holón Controlador (HC)</b>	C	<ul style="list-style-type: none"> <li>Realiza el control de las tasas de servicio de los holones servidores (HS).</li> </ul>
<b>Holón Servidor (HS)</b>	H/C	<ul style="list-style-type: none"> <li>Representa la parte física y lógica de un servidor dentro del sistema de colas, este Holón, envía su status operativo al HL, HR, HC.</li> </ul>
<b>Holón de Línea (HL)</b>	H	<ul style="list-style-type: none"> <li>Supervisa a todos los HS pertenecientes a una línea de producción y toma el control de las tasas de servicio de los HR en caso de un aumento/disminución de esta producto de una falla.</li> <li>Reenvía status de falla y cantidad de entidades en cola de cada HS que supervisa.</li> </ul>
<b>Holón Ruteo (HR)</b>	H	<ul style="list-style-type: none"> <li>Supervisa el status de falla de cada HS dentro de su radio.</li> <li>En caso de falla de uno de los HS, el HR toma el control del ruteo, transformando el ruteo equiprobable en un ruteo inteligente.</li> </ul>

C: Aplica en Arquitectura Centralizada; H: Aplica en Arquitectura Heterárquica

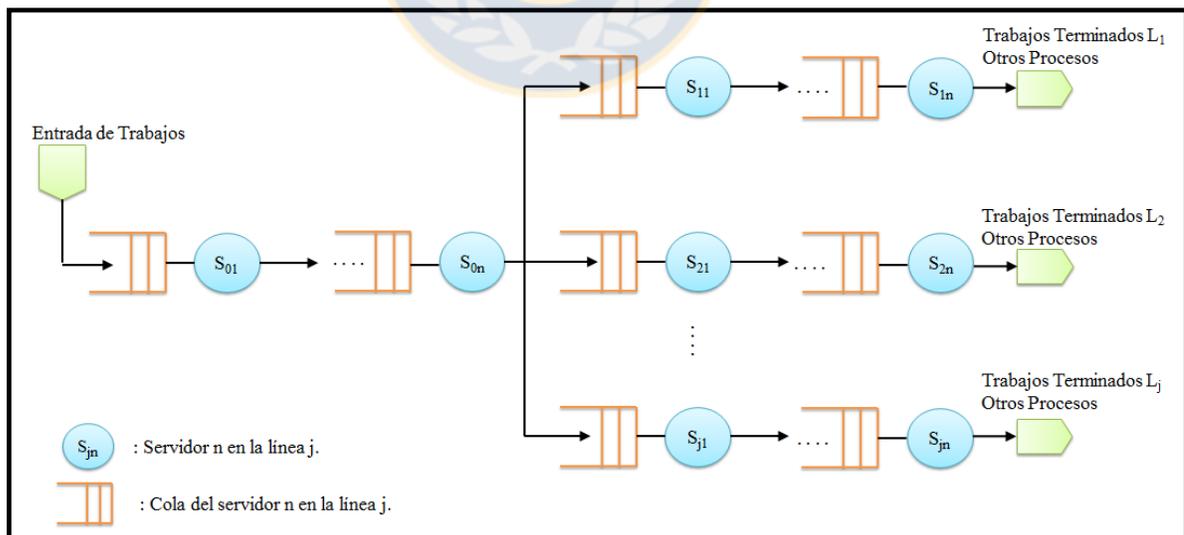
Considerando los conceptos de redes de cola definidos en [39], la aplicación de la arquitectura propuesta puede ser realizada en todo tipo red de cola, es decir, con cualquier distribución de probabilidad en intervalos entre arribo, tasas de servicio y disciplina de cola. Sin embargo, el algoritmo de control de cada holón de la arquitectura es dependiente del tipo de red de cola a controlar.

### 3.2.1 Algoritmo de control de Arquitectura Heterárquica

El algoritmo de control desarrollado se limita a procesos que pueden ser representados como una red cola de Jackson (definida en [39]). A continuación se resume las principales características de este tipo de red:

- Existe solo una clase de trabajo.
- El número de trabajos en la red es ilimitado.
- Cada nodo en la red puede tener un arribo con distribución de probabilidad Poisson desde afuera de la red.
- Todas las tasas de servicio son en base a una distribución de probabilidad Exponencial.
- La disciplina en cada nodo es FCFS (First-Come-First-Served)
- Cada nodo consiste en  $m_i \geq 1$  estaciones de servicio idénticas con la tasa de servicio  $\mu_i$ ,  $i=1, \dots, N$ . La tasa de arribo  $\lambda_{0i}$ , como las tasas de servicio pueden depender del número de  $K_i$  trabajos en el nodo. En este caso se tiene una tasa de servicio dependiente de la carga y una tasa de arribo dependiente de la carga.

En particular, para el algoritmo propuesto se asumirá una red de cola según notación la Kendall M/M/1-FCFS. Gráficamente se observa la implementación de la red en la Figura 16.



**Figura 16:** Red de Cola aplicable al algoritmo de control propuesto.  
Fuente: Elaboración propia.

En la Figura 17 se muestra a través de diagramas de flujo los algoritmos embebidos en el holón de Ruteo y Holón de Línea y describen a continuación:

**Holón de Ruteo (HR):** está constantemente monitoreando el status operacional de cada holón servidor, cuando este detecta una falla, realiza un ordenamiento de colas de línea en función de largo de las mismas. Luego, se asigna la probabilidad más baja de que un trabajo se dirija a esa cola a aquella línea con la cola más larga (ya que no se desea saturar más esta línea) y se aumenta proporcionalmente la probabilidad hasta asignar la más alta probabilidad a la línea con la cola más corta. Lo anterior produce una distribución de los trabajos de manera uniforme.

**Holón de Línea (HL):** al igual que el HR el Holón de Línea monitorea el status operacional de cada holón servidor y en caso de falla, toma el control de las tasas de servicio de los holones servidores de su línea, en caso de que la falla provenga de un holón servidor que pertenece a su línea, el holón de línea disminuye a la tasa de falla la tasa de servicio del resto de los holones servidores de su línea con el fin de mantener el equilibrio y disminuir la acumulación de trabajo en colas intermedias.

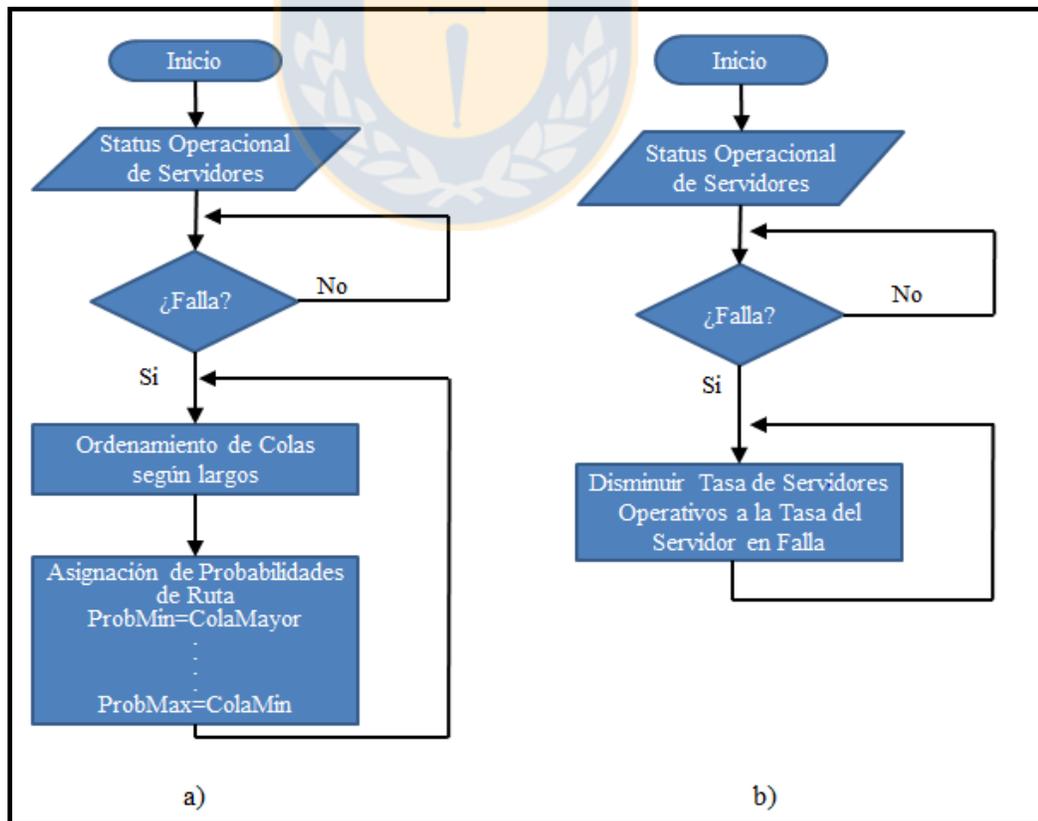


Figura 17: a) Diagrama de Flujo Holón Ruteo (HR). b) Diagrama de Flujo Holón de Línea (HL). Fuente: Elaboración propia.

### 3.2.2 Limitaciones del sistema de control propuesto

El sistema de control propuesto cuenta con las siguientes limitaciones:

1. El sistema está diseñado para controlar Redes de Jackson, por lo tanto, solo acepta una sola clase de trabajo.
2. No existen líneas de producción dedicadas a un tipo de trabajo en específico, por lo tanto, el sistema está diseñado para enviar un trabajo a cualquier línea.
3. Se limita a provocar fallas aguas arriba del punto de ruteo.

### 3.2.3 Resultados Esperados

Los resultados esperados producto de la aplicación de CRC-H son los siguientes:

1. Disminución de la cola de trabajo de la línea en falla. El algoritmo disminuye la probabilidad de que un trabajo se dirija hacia la línea con la cola más larga, que en situación de falla sería la línea en falla, y aumenta la probabilidad que un trabajo se dirija hacia las líneas sin falla, por lo tanto, existe una redistribución de los trabajos en función del largo de las colas de cada línea.
2. Estabilización del nivel de utilización de los servidores aguas arriba del servidor en falla. Una falla en un servidor se considera como una disminución en su tasa de servicio y dado que los servidores aguas arriba están sin falla, la utilización de estos últimos baja abruptamente puesto que la tasa de arribo de trabajos hacia su cola disminuye. Ante una falla, el algoritmo iguala la tasa de servicio de los servidores sin falla a la tasa del servidor en falla, por lo tanto, la utilización se mantiene estable independientemente de que exista una falla o no.

### 3.3 Método de Experimentación

Con el fin de verificar la hipótesis planteada se procederá a validar el sistema de control propuesto a través de simulación. Para ello, se propone la siguiente metodología de experimentación:

1. Definir el proceso de red de cola a simular.
2. Definir la Holarquía del CRC-H.
3. Desarrollar una plataforma para simular y controlar una red de cola.
4. Experimentación.

#### 3.3.1 Definición del Proceso a Simular

El proceso a simular consiste en una red de Jackson con una línea común ( $L0$ ) y tres líneas paralelas ( $L1$ ,  $L2$  y  $L3$ ) con los parámetros indicados en la Tabla 5, mientras que la implementación física se muestra en la Figura 18.

Tabla 5: Parámetro de red de cola a simular

Parámetro	Valor/Tipo
Servidores en $L0$	2
Servidores en $L1$	2
Servidores en $L2$	2
Servidores en $L3$	2
Distribución de Probabilidad de arribo	Poisson
Distribución de Probabilidad de Tasa de Servicio	Exponencial
Media de Tasa de Arribo de Trabajos ( $\lambda_{0i}$ )	$5 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$
Media de Tasa de Servicio $L0$ ( $\mu_i$ )	$5 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$
Media de Tasa de Servicio $L1$ , $L2$ y $L3$ ( $\mu_i$ )	$15 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$
Probabilidad de Ruteo hacia $L1$ , $L2$ y $L3$	Equiprobable

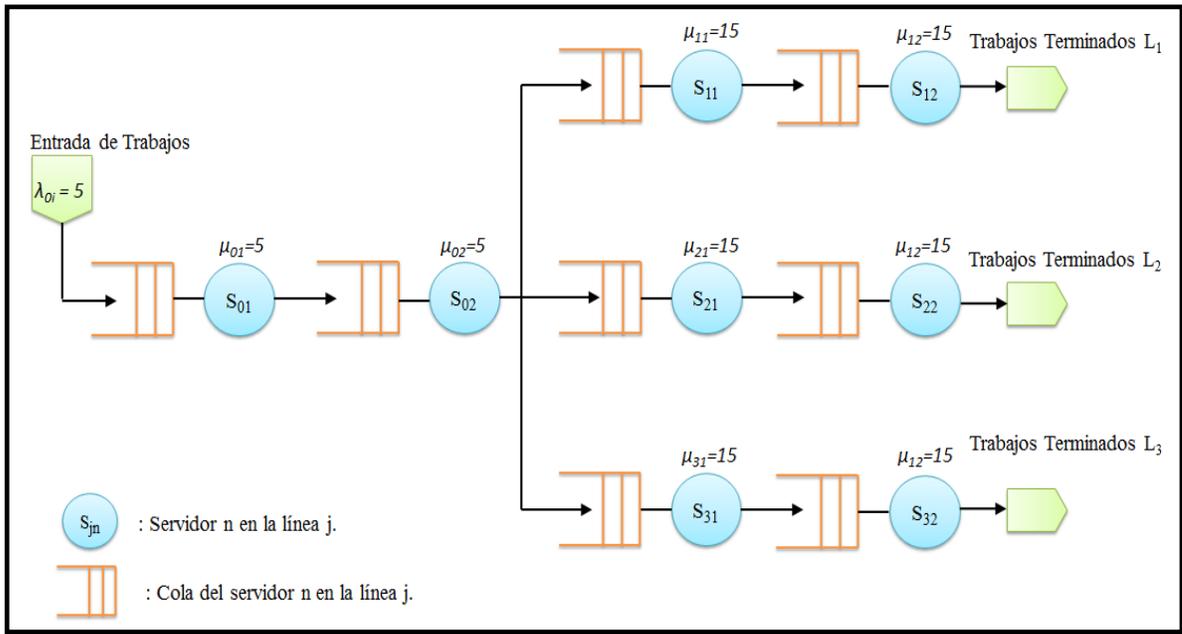


Figura 18: Proceso de experimentación



### 3.3.2 Definición de la Holarquía del CRC-H

La Holarquía para el proceso especificado en el punto 0 según la arquitectura propuesta se detalla en la Tabla 6 y considera:

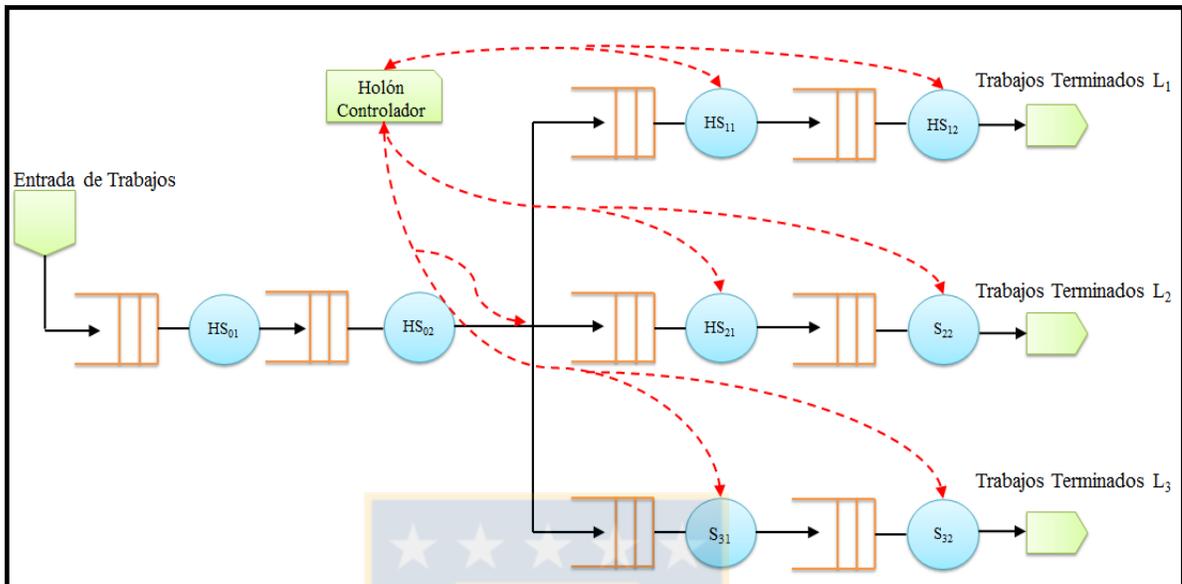
- ✓ (1) holón controlador el cual controla las tasas de servicio de todos los holones servidores cuando la arquitectura está funcionando bajo control centralizado.
- ✓ (8) holones servidores que representan los servidores del proceso.
- ✓ (4) holones de línea que toman el control de las tasas de servicio de los holones servidores de su respectiva línea cuando la arquitectura funciona bajo control distribuido.
- ✓ (1) Holón de Ruteo el cual toma el control del ruteo de trabajos en función de las colas de los servidores  $S_{IN}$  con  $N=1...3$ , cuando la arquitectura funciona bajo control distribuido.

Tabla 6: Definición de Holones

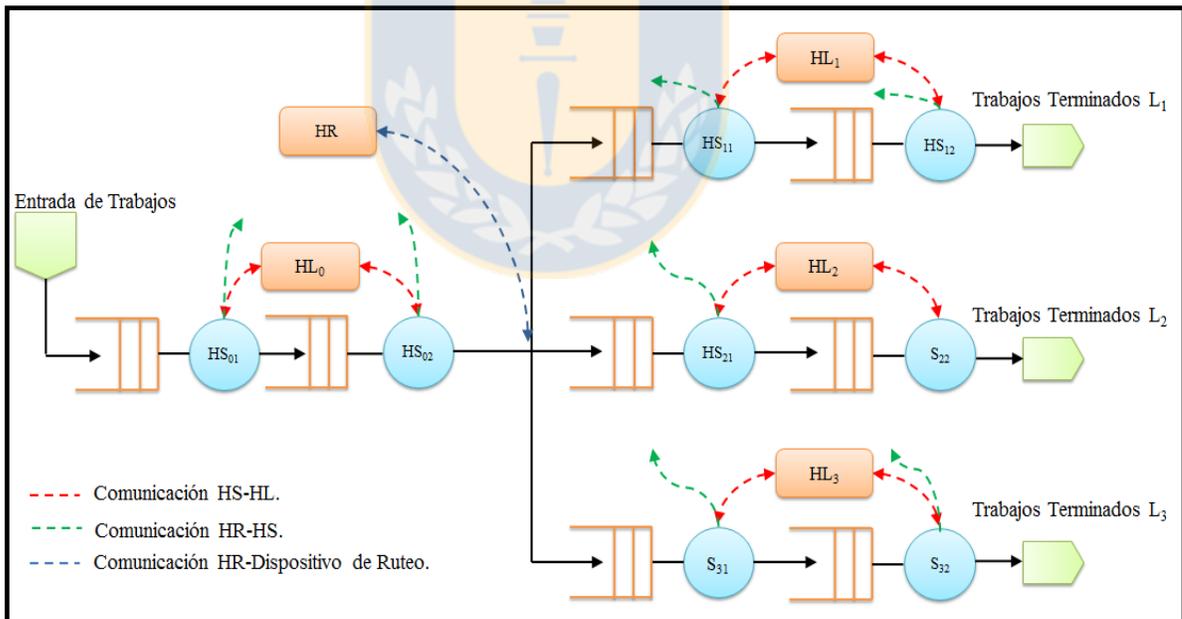
Entidad	Descripción
<b>Holón Controlador (HC)</b>	<ul style="list-style-type: none"> <li>• <i>HC</i>: Controla el ruteo de trabajos y las tasas de servicio de los holones servidores (HS).</li> </ul>
<b>Holón Servidor (HS)</b>	<ul style="list-style-type: none"> <li>• <i>HS<sub>01</sub></i>: Holón asociado al Servidor 1 de la Línea 0.</li> <li>• <i>HS<sub>02</sub></i>: Holón asociado al Servidor 2 de la Línea 0.</li> <li>• <i>HS<sub>11</sub></i>: Holón asociado al Servidor 1 de la Línea 1.</li> <li>• <i>HS<sub>12</sub></i>: Holón asociado al Servidor 2 de la Línea 1.</li> <li>• <i>HS<sub>21</sub></i>: Holón asociado al Servidor 1 de la Línea 2.</li> <li>• <i>HS<sub>22</sub></i>: Holón asociado al Servidor 2 de la Línea 2.</li> <li>• <i>HS<sub>31</sub></i>: Holón asociado al Servidor 1 de la Línea 3.</li> <li>• <i>HS<sub>32</sub></i>: Holón asociado al Servidor 2 de la Línea 3.</li> </ul>
<b>Holón de Línea (HL)</b>	<ul style="list-style-type: none"> <li>• <i>HL<sub>0</sub></i>: Holón de Línea asociado a la Línea 0.</li> <li>• <i>HL<sub>1</sub></i>: Holón de Línea asociado a la Línea 1.</li> <li>• <i>HL<sub>2</sub></i>: Holón de Línea asociado a la Línea 2.</li> <li>• <i>HL<sub>3</sub></i>: Holón de Línea asociado a la Línea 3.</li> </ul>
<b>Holón Ruteo (HR)</b>	<ul style="list-style-type: none"> <li>• <i>HR</i>: Ejecuta el control inteligente en caso de falla de algún HS.</li> </ul>

C: Aplica en Arquitectura Centralizada; H: Aplica en Arquitectura Heterárquica

En la Figura 19 y Figura 20 se muestra gráficamente la interacciones entre los holones que componen la holarquía definida anteriormente tanto en la arquitectura centralizada (sistema sin falla) y Distribuida-Heterárquica (con falla) respectivamente.



**Figura 19: Holarquía en Arquitectura Centralizada (Proceso sin Falla)**



**Figura 20: Holarquía en Arquitectura Descentralizada-Heterárquica (Proceso con Falla)**

### **3.3.3 Diseño de la Plataforma para simular y controlar una red de cola**

Las etapas del diseño de la plataforma de pruebas son las siguientes:

1. Selección del Software y Hardware a utilizar.
2. Modelar la red de cola definida en el punto 3.6 y el sistema de control en la plataforma de simulación.
3. Diseñar y programar los algoritmos de control.
4. Diseñar los GUI de cada tipo de Holón.
5. Planificar el proceso de experimentación.

#### **3.3.3.1 Selección del Software y Hardware**

La plataforma diseñada consiste en utilizar como herramienta de simulación el entorno gráfico de programación Simulink de Matlab con el apoyo del toolbox SimEvents el cual proporciona las librerías para simular procesos a eventos discretos. Dado que Matlab no cuenta con una plataforma que soporte el paradigma holónico, se hace necesario considerar un software en donde los holones puedan ser ejecutados. Para ello, dado que se comprueba en trabajos anteriores [6] que la tecnología multiagentes es ideal para la implementación de sistemas holónicos, además de las ventajas indicadas en el capítulo 4, la amplia información disponible y la preexistencia de un middleware que permite la comunicación bidireccional con Matlab llamado MACSimJX (Ver descripción detallada de este software en el Anexo N°2), el software seleccionado para este fin es JADE. En la Tabla 7 se resume el nombre del software a utilizar, versión y un breve descriptivo, mientras que en la Tabla 8 se lista el hardware utilizado para levantar la plataforma de pruebas.

Tabla 7: Software necesario para la experimentación

Función	Software
<b>Herramienta de Simulación</b>	<ul style="list-style-type: none"> <li>• <b>Matlab 2010a</b>, 32 bits (Abreviatura de <b>MATrix LABORatory</b>). es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE). Para simular la red de cola se utilizarán las siguientes extensiones de Matlab:               <ul style="list-style-type: none"> <li>✓ <b>Simulink</b>. Entorno de diagramas de bloque para la simulación multidominio y el diseño basado en modelos en Matlab. Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y solvers para modelar y simular sistemas dinámicos.</li> <li>✓ <b>SimEvents</b>. Toolbox que extiende las posibilidades de simulink para aplicar herramientas para simulación a eventos discretos. Contiene librerías con bloques predefinidos tales como colas, servidores, ruteadores, etc.</li> </ul> </li> </ul>
<b>Entorno de Ejecución de Holones</b>	<ul style="list-style-type: none"> <li>• <b>JADE 3.6.1</b>. (Abreviatura de JAVA Agent Developmet Framework). Plataforma para el desarrollo de agentes, implementada en Java. JADE soporta la coordinación de múltiples agentes FIPA y proporciona una implementación estándar del lenguaje de comunicación FIPA-ACL, la cual facilita la comunicación entre agentes y permite la detección de servicios que se proporcionan en el sistema.</li> </ul>
<b>Comunicación holones/Simulación</b>	<ul style="list-style-type: none"> <li>• <b>MACSimJX para Matlab 2010</b> (Abreviatura de <b>Multi-Agent Control for Simulink</b>). Este software crea un túnel de comunicación entre JADE y Matlab para intercambiar datos entre las entidades de control y el proceso.</li> </ul>
<b>Programación de Holones</b>	<ul style="list-style-type: none"> <li>• <b>Netbeans IDE 8.0.2</b>. Entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Con el fin de facilitar la depuración del código de los holones desarrollados se utiliza este entorno.</li> </ul>

**Tabla 8: Hardware necesario para la experimentación**

<b>Equipo</b>	<b>Características</b>
<b>Computador Portátil</b>	<ul style="list-style-type: none"><li>• Sistema Operativo: Windows XP, S.P. 3, 32bits.</li><li>• Procesador : Intel Core i5 @ 2.60Ghz</li><li>• Memoria RAM : 2.8GB.</li></ul>

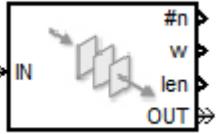
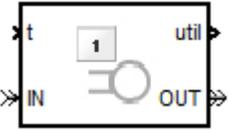
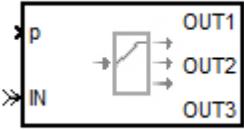
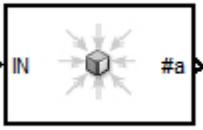
Cabe resaltar que la limitación a utilizar software de 32bits es dada por el software MACSimJX, el cual en su estado actual de desarrollo es soportado en tal arquitectura.



### 3.3.3.2 Modelación de la red de cola en la plataforma de simulación

Como se definió en el punto anterior, la modelación y simulación del proceso seleccionado será implementada en Simulink de Matlab utilizando las librerías del toolbox SimEvents. En la Tabla 9, se describe cada bloque utilizado para modelar y controlar el proceso trasapado a Simulink el cual se muestra en la Figura 21.

Tabla 9: Bloques del Toolbox Simevents para la aplicación

Bloque	Descripción del bloque
 <p><b>FIFO Queue</b></p>	<p>Almacena hasta <math>N</math> entidades, las cuales salen del bloque con disciplina <i>FIFO</i> (First-in, First-Out).</p> <p><b>#n:</b> Número de entidades que actualmente están en cola.</p> <p><b>w:</b> Tiempo de espera promedio en el bloque por las entidades.</p> <p><b>len:</b> Número promedio de entidades en cola.</p>
 <p><b>Single Server</b></p>	<p>Sirve a una entidad por un periodo de tiempo.</p> <p><b>t:</b> Tiempo de servicio definido externamente para una nueva entidad.</p> <p><b>Útil:</b> Utilización del servidor.</p>
 <p><b>Output Switch</b></p>	<p>Selecciona un puerto de salida para liberar una entidad.</p> <p><b>p:</b> Puerto que permite a través una señal externa seleccionar la ruta (puerto de salida) de la entidad.</p>
 <p><b>Time-Based Entity Generator</b></p>	<p>Genera entidades usando una generación de tiempo desde una señal o distribución estadística.</p> <p><b>t:</b> Puerto que permite a través una señal externa definir el tiempo entre arribo de entidades.</p> <p><b>d:</b> Número de entidades que salen del bloque.</p>
 <p><b>Entity Sink</b></p>	<p>Acepta o bloquea entidades. Este bloque provee una forma de terminar la ruta de una entidad.</p>

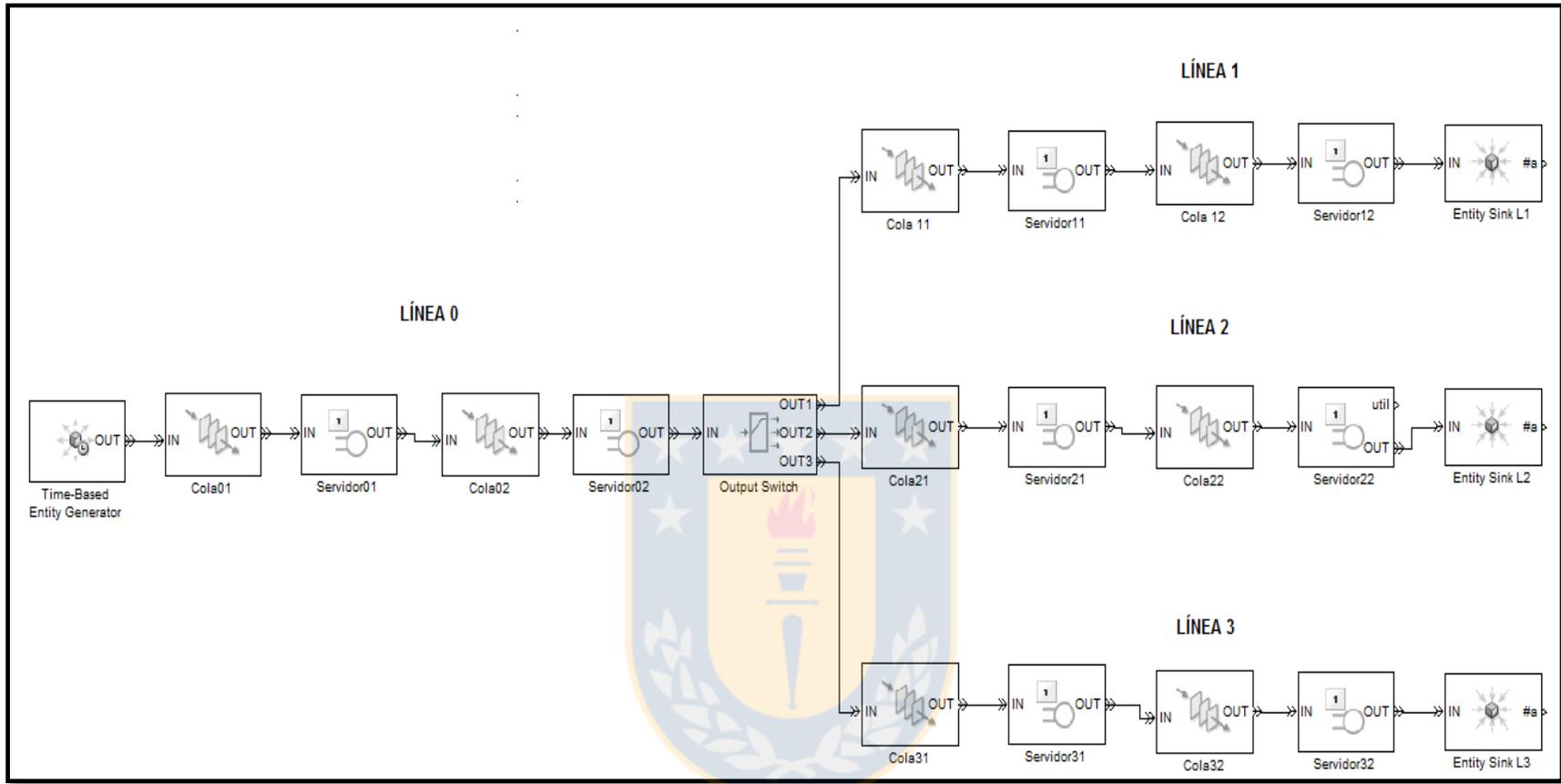


Figura 21: Proceso Implementado en Simulink

Luego de tener implementado el proceso en Simulink, se utiliza la función “S-function” que se incluye en el paquete de descarga del software MACSimJX (para mayores detalles de instalación y configuración ver el Anexo 2) el cual permite la comunicación entre Simulink y JADE en tiempo real. Este bloque cuenta con un bloque multiplexor que contiene todos los puertos de entrada a los cuales se conectan los datos a transmitir hacia JADE, este bloque combina los datos recibidos en los puertos y los envía a través de un solo vector. Por otro lado, cuenta con un bloque demultiplexor, el cual extrae los datos contenidos en un vector enviado desde JADE y los separa en distintos puertos de salida. En la Tabla 10 se muestran las asignaciones de señales a los puertos de entradas y salidas del S-Function para la experimentación mientras que en la Figura 22, se muestra el proceso implementado en Simulink y comunicado al entorno de ejecución de los holones JADE a través de MACSimJX.

Con el fin de comparar las variables asociadas a las colas y servidores de la red, se conectan a cada señal un bloque “Scope” de Simulink el cual permite almacenar los valores en una matriz para posteriormente comparar y graficar las variables en los escenarios descritos en la experimentación (ver punto 3.3.4).

Tabla 10: Interfaz Entrada/Salida entre Simulink/JADE

<b>Puerto de Entrada</b>	<b>Señal</b>	<b>Puerto de Salida</b>	<b>Señal</b>
1	Cola de $S_{01}$ .	1	Ruta
2	Cola de $S_{02}$ .	2	Probabilidad de Ruta $L1$
3	Cola de $S_{11}$ .	3	Probabilidad de Ruta $L2$
4	Cola de $S_{12}$ .	4	Probabilidad de Ruta $L3$
5	Cola de $S_{21}$ .	5	Tiempo de Servicio de $S_{01}$ .
6	Cola de $S_{22}$ .	6	Tiempo de Servicio de $S_{02}$ .
7	Cola de $S_{31}$ .	7	Tiempo de Servicio de $S_{11}$ .
8	Cola de $S_{32}$ .	8	Tiempo de Servicio de $S_{12}$ .
9	Tiempo de Simulación	9	Tiempo de Servicio de $S_{21}$ .
		10	Tiempo de Servicio de $S_{22}$ .
		11	Tiempo de Servicio de $S_{31}$ .
		12	Tiempo de Servicio de $S_{32}$ .

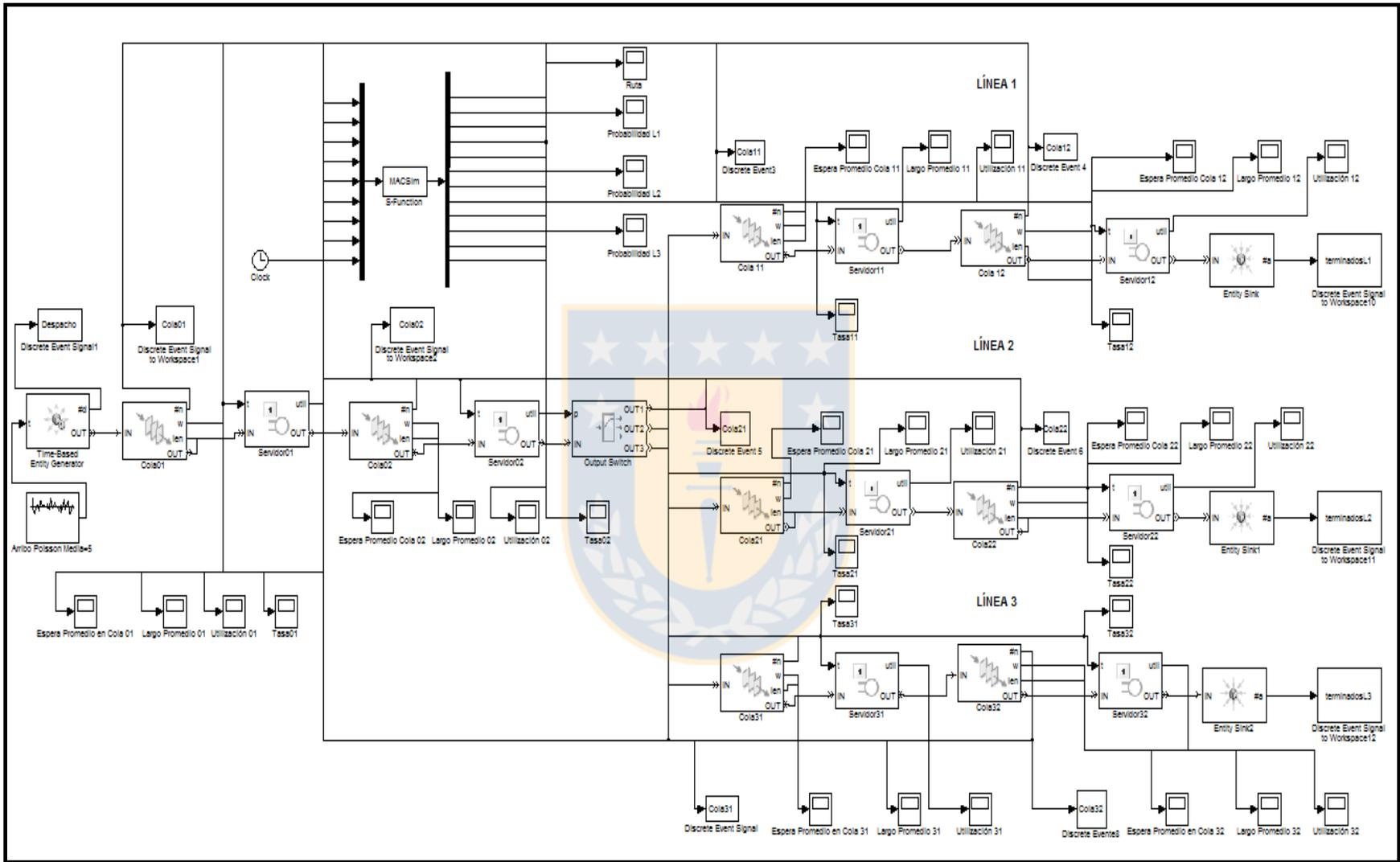


Figura 22: Proceso Controlado y Supervisado por CRC-H

### 3.3.3.3 Diseño de Holones

El diseño de cada holón consiste en dos partes:

- ✓ Diseño de la lógica de control.
- ✓ Interfaz de usuario (GUI).

#### 3.3.3.3.1 Consideraciones de diseño

Para el diseño de la lógica se tomarán las siguientes consideraciones:

- La arquitectura centralizada será representada por la red de cola en equilibrio, es decir, bajo las condiciones indicadas en el punto 0. Dado lo anterior, el holón HC no será programado y la programación se concentrará en el algoritmo de la arquitectura descentralizada.
- Los holones HS serán representados por los bloques servidores de Simulink, por lo tanto, la transmisión de los estatus de cola de los servidores hacia el holón HR serán a través mensajes enviados por los holones de línea.
- La base de programación de MACSimJX no está diseñada para enviar mensajes a un holón puntual sino que se realiza un envío broadcast, por lo tanto los envíos de información específicos de la arquitectura CRC-H será implementado a través del filtrado del mensaje.

#### 3.3.3.3.2 Diseño de algoritmos de control

Bajo las condiciones anteriores, los servidores reportan su cola y status de falla tanto a los holones de línea (HL) como al holón de ruteo (HR), cuando no existe falla las tasas se mantienen en equilibrio, por lo tanto, según lo indicado anteriormente se considerará que el sistema está en modo centralizado (en el rendimiento óptimo de la red). Cuando existe una falla de un servidor, se activan los algoritmos de control de la arquitectura distribuida:

1. EL HL controla las tasas de los servidores de su línea. Detectada la falla, el HL disminuye a la tasa de falla, la tasa de servicio de todos los servidores sin falla.
2. El HR controla el ruteo de trabajos. Detectada la falla, el HR compara el largo de cola del primer servidor en cada línea aguas arriba al punto de ruteo y asigna la probabilidad más alta, una probabilidad media y una probabilidad baja a la cola más corta, cola media y cola más larga respectivamente. Para este proceso en particular, se

han definido las probabilidades indicadas en la Tabla 11, sin embargo, esto puede variar dependiendo de cada proceso en particular, la probabilidad asignada modifica la sensibilidad de crecimiento de cada cola. En la Figura 23 se muestra el diagrama de flujo asociado a la interacción entre los holones de la holarquía.

**Tabla 11: Regla de asignación de probabilidades de HR**

Largo de Cola	Probabilidad	Valor
Más corta	Máxima (ProbMax)	0.7
Media	Media (ProbMed)	0.2
Más larga	Min (ProbMin)	0.1

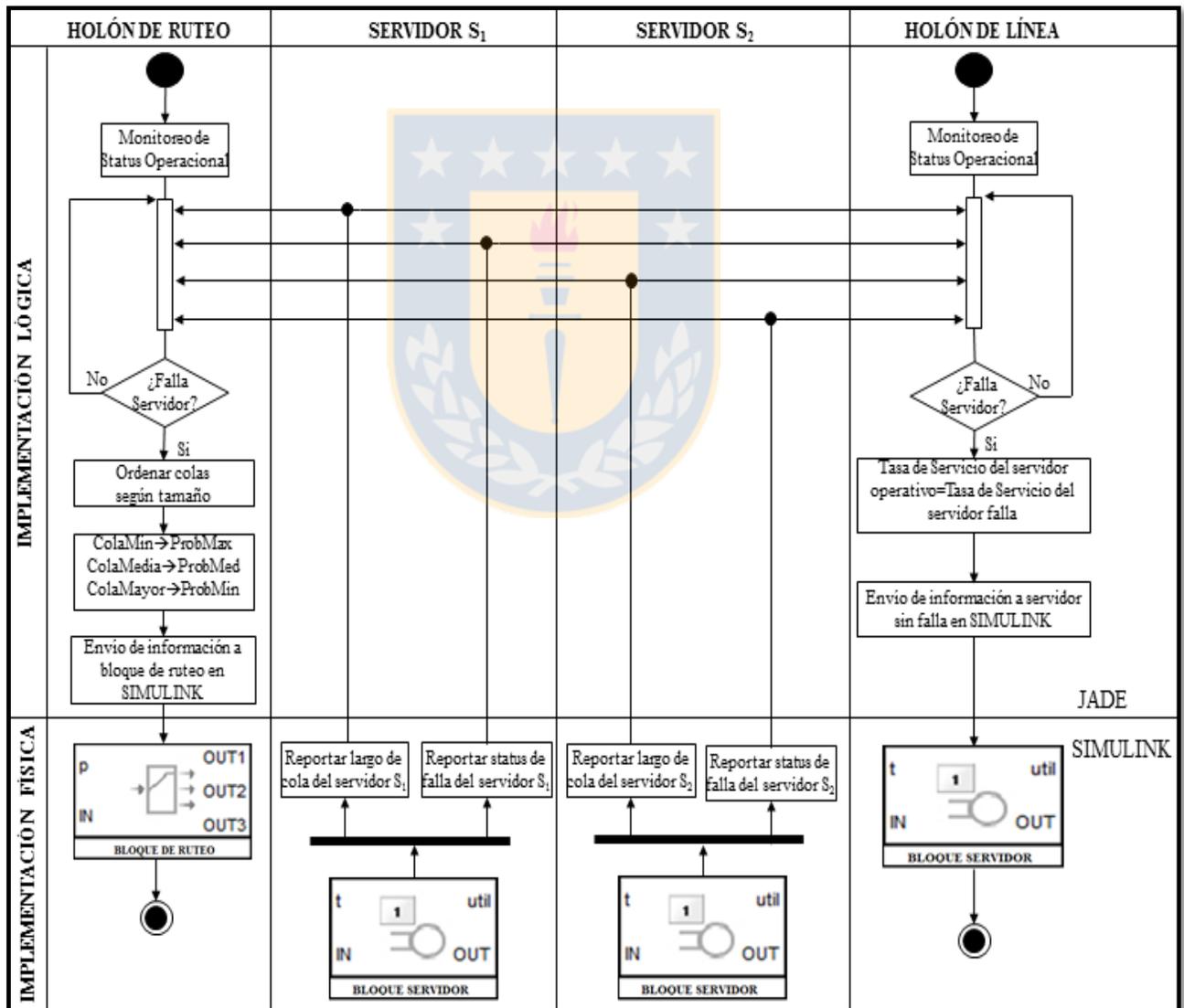


Figura 23: Diagrama de Flujo del algoritmo de control para Holón de Ruteo y Holón de Línea

### 3.3.3.3 Diseño de Interfaz de Usuario (GUI) de cada tipo de Holón

Cada holón tendrá una GUI la cual estará habilitada para interactuar con el usuario.

Para el Holón de Ruteo se define una GUI con la cual el usuario podrá visualizar:

1. Tipo de Ruteo: Indica si el sistema está funcionando con un ruteo equiprobable o inteligente, este último caso se refiere al ruteo cuando el sistema actúa según una arquitectura de control distribuido (heterárquico).
2. Estado del Sistema: Indica si el sistema está operativo o en falla.
3. Estado de las Colas: Indica cual es la cola más larga y corta en un tiempo determinado. (solo habilitada para debug del programa).
4. Tiempo de Simulación: Indica el tiempo actual de simulación.
5. Recuperación: Indica si está habilitada en la simulación la opción de recuperación de falla.

En la Figura 24 se muestra el diseño de la GUI del holón de ruteo en forma gráfica.

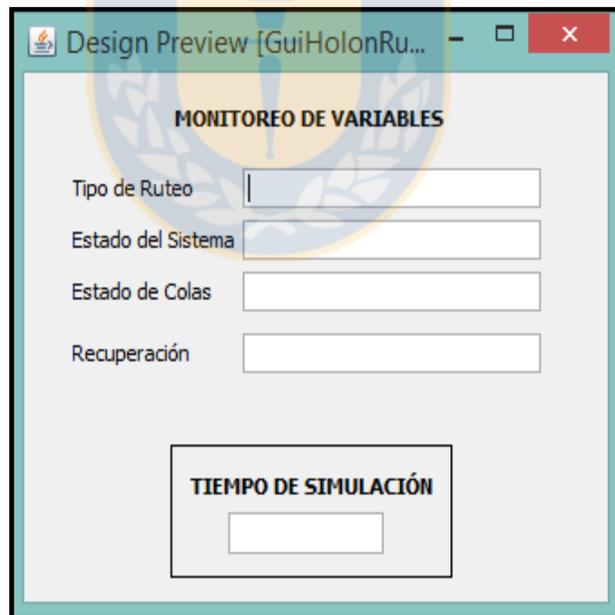


Figura 24: Layout del GUI diseñado para el Holón de Ruteo

Para el Holón de Línea se define una GUI con la cual el usuario podrá visualizar:

1. Media de Tasa de Servicio de los servidores de la línea: se muestra la Media de la tasa de servicio actual.
2. Status de Servidores: indica si los servidores están en un estado operativo o en falla y lo individualiza.
3. Simulación de Falla: el usuario puede simular una falla, además, puede seleccionar si se simula una recuperación de falla o no. Ambos parámetros son definidos en un tiempo  $t$  modificable por el usuario en la programación del holón de línea.
4. Ingreso del valor medio de la tasa de servicio de falla: se ingresa el valor de la media de las tasas de servicio a la cual el usuario desea simular la falla. Presionando el botón de “Actualizar” la tasa de falla se actualiza en el algoritmo del holón de línea. Presionando el botón de “Reset de Falla”, la tasa de servicio se reestablece a los valores originales.

En la Figura 25 se muestra el diseño de la GUI del holón de línea en forma gráfica.



Figura 25: Layout del GUI diseñado para el Holón de Línea

### 3.3.4 Procedimiento de Experimentación

Con el fin de evaluar métricas de desempeño de un control convencional respecto del control propuesto (CRC-H) se define como experimentación la comparación de métricas de desempeño en los escenarios detallados en la Tabla 12.

Tabla 12: Definición de escenarios de experimentación

Escenario	Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla $\left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	Tiempo de ejecución (Te) y recuperación (Tr) de falla $[\text{Unidad de Tiempo}]$
1	$S_{11}$	Sin recuperación	45	Te = 400
2	$S_{11}$	Sin recuperación	90	Te = 400
3	$S_{21}$	Sin recuperación	45	Te = 400
4	$S_{21}$	Sin recuperación	90	Te = 400
5	$S_{31}$	Sin recuperación	45	Te = 400
6	$S_{31}$	Sin recuperación	90	Te = 400
7	$S_{11}$	Con recuperación	45	Te = 400 Tr = 1200
8	$S_{21}$	Con recuperación	45	Te = 400 Tr = 1200
9	$S_{31}$	Con recuperación	45	Te = 400 Tr = 1200

Las métricas a comparar serán las siguientes:

1. Utilización de todos los servidores.
2. Cola Instantánea de cada servidor aguas arriba al punto de ruteo.
3. Trabajos terminados.

Consideraciones:

- Tiempo de Simulación=2000 unidades de tiempo.
- Se asume que el control convencional consiste en el proceso con los parámetros en equilibrio, es decir, los indicados en el punto 0.
- Luego de una falla, el control convencional no realiza ninguna acción correctiva, por lo tanto, el ruteo sigue siendo equiprobable.
- Se simularán fallas solo en el servidor 1 de una línea ya que una falla en el servidor 2 produce la misma acción de control y por lo tanto gráficas similares.

# CAPÍTULO 4

## Resultados

### 4.1 Introducción al capítulo

Luego de definir los métodos de experimentación y materiales a utilizar en el capítulo 3, el presente capítulo muestra los resultados obtenidos en las simulaciones realizadas para cada escenario planteado. Para el detalle de todas las gráficas obtenidas como resultado en los distintos escenarios ver Anexo 3.

### 4.2 Escenario base: Control Convencional sin Falla

En primer lugar, se tomará como escenario base al proceso controlado de forma convencional, es decir, bajo las condiciones de equilibrio y sin falla, la performance de las métricas obtenidas en el escenario base será comparada con el sistema convencional en falla y CRC-H en falla bajo los distintos escenarios definidos.

En la Tabla 13 se muestra un resumen con los resultados luego de simulación. Considerando que el escenario base es la mejor situación posible, se tiene entonces que la máxima cantidad de trabajos terminados es de 346 trabajos y la suma de la utilización de cada servidor de las líneas *L1*, *L2* y *L3* es de 5,18 como máximo y 4,98 al finalizar la simulación.

**Tabla 13: Resultados de Métricas Escenario Base**

Línea	Servidor	Control Escenario Base				Trabajos Terminados
		Trabajos en cola		Utilización		
		Máx	Final	Máx	Final	
1	1	12	4	0,85	0,79	116
	2	7	2	0,85	0,72	
2	1	11	0	0,82	0,82	117
	2	9	3	0,83	0,82	
3	1	19	10	0,92	0,92	113
	2	12	5	0,91	0,91	
Total				5,18	4,98	346

### 4.3 Escenario 1: Falla en el servidor $S_{11}$ , sin recuperación

En la Tabla 14 se resumen los parámetros asociados al escenario 1.

Tabla 14: Parámetros del escenario 1

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución ( $T_e$ ) y recuperación ( $T_r$ ) de falla
$S_{11}$	Sin recuperación	$45 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$T_e = 400$ [Unidad de Tiempo]

En la Tabla 15 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 16 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 1 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 1, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en cola de 31,43%, 33,33% y 9,52% respectivamente en relación al escenario base. Lo anterior no ocurre en el control convencional en el cual la máxima cantidad de trabajos en cola del servidor  $S_{11}$  aumenta significativamente de 12 a 65 (+68,83%), mientras que el resto de las líneas se mantienen con un rendimiento idéntico respecto al escenario base.

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,93 este valor es incluso mayor respecto al escenario base que tiene una utilización máxima de 0,85, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 4,33%. Al final de la simulación la utilización del servidor sin falla es de 0,93, valor que es mayor al escenario base que tiene una utilización de 0,72, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 12,52%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{12}$  de 0,85 a 0,93, lo cual se traduce en una disminución del 4,33% en el tiempo de ocio servidor  $S_{12}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{12}$  de 0,45 a 0,93 respecto al control convencional, lo que se traduce en una disminución del 34,88% en el tiempo de ocio servidor

$S_{12}$ , ver Figura 25. La baja utilización del servidor  $S_{12}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{12}$  provocada por la falla en el servidor  $S_{11}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 2,51% y aumenta la utilización final total en 6,6% respecto del control convencional bajo las condiciones del escenario 1.

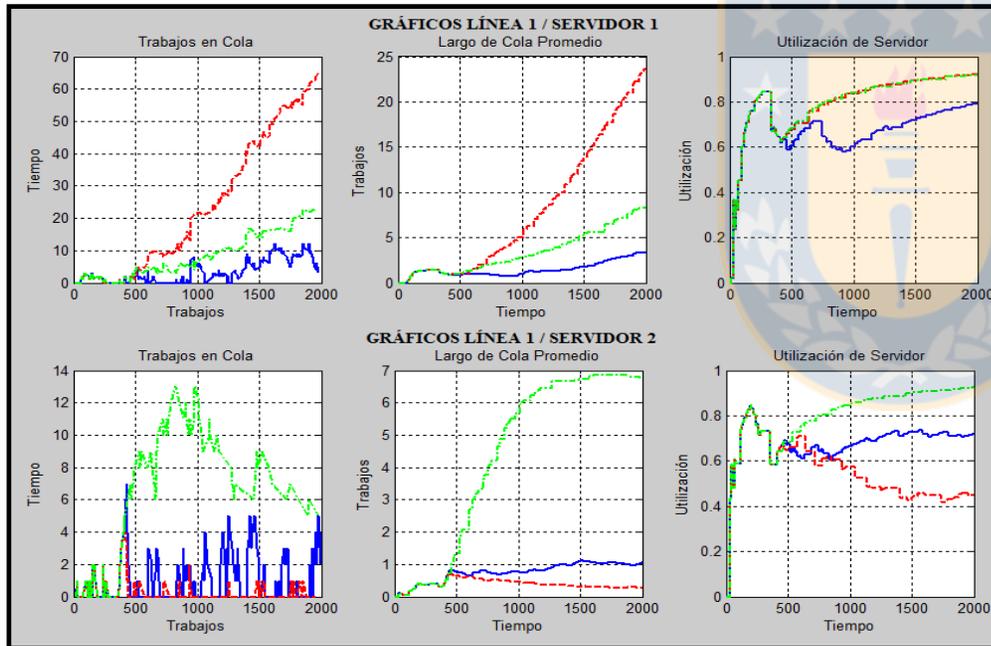
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{11}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 116 a 50 equivalente a un -39,76%, mientras que el control convencional disminuye de 116 a 58 equivalente a un -33,33%. Lo anterior indica que CRC-H produce ocho trabajos menos que el control convencional, es decir, CRC-H disminuye en 7,41% los trabajos terminados en la línea en falla respecto al control convencional. Sin embargo, se observa una mayor producción en  $L2$  respecto al escenario base (+4,49%) y al control convencional (+4,49%), ver Figura 26, lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 288 a 291 equivalente a +0,52% respecto del control convencional bajo las condiciones del escenario 1.

**Tabla 15: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 1**

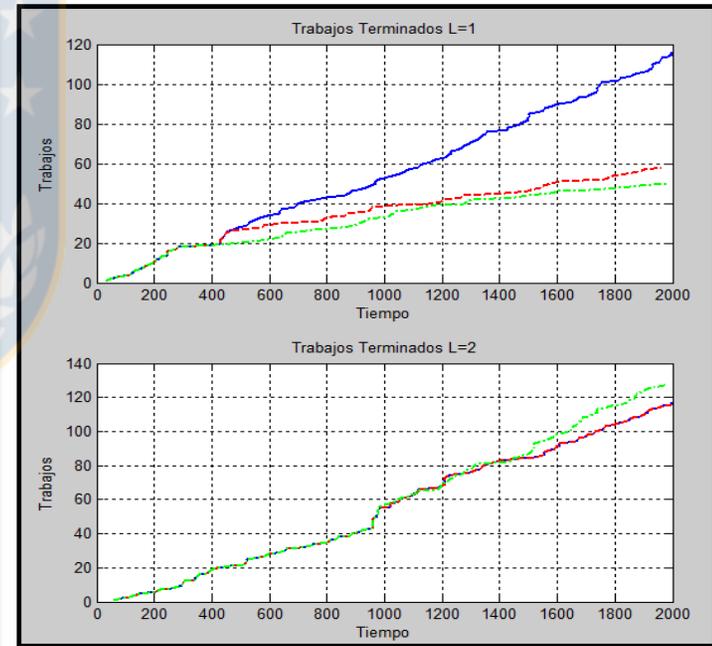
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	65	65	0,92	0,92	58	23	23	0,92	0,92	50
	2	5	0	0,85	0,45		13	5	0,93	0,93	
2	1	11	0	0,82	0,82	117	22	17	0,94	0,94	128
	2	9	3	0,83	0,82		18	9	0,92	0,92	
3	1	19	10	0,92	0,92	113	23	19	0,92	0,92	113
	2	10	5	0,91	0,91		10	5	0,91	0,91	
Total				5,26	4,84	288			5,53	5,53	291

**Tabla 16: Comparación porcentual entre controles, Escenario 1**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final			
1	1	68,83	88,41	4,26	7,53	-33,33	31,43	70,37	4,16	7,43	-39,76	-47,73	-47,73	-0,10	-0,10	-7,41
	2	-16,67	-100,00	0,00	-23,38		30,00	42,86	4,33	12,52		44,44	100,00	4,33	34,88	
2	1	0,00	0,00	0,00	0,00	0,00	33,33	100,00	6,38	6,81	4,49	33,33	100,00	6,38	6,81	4,49
	2	0,00	0,00	0,00	0,00		33,33	50,00	4,77	5,40		33,33	50,00	4,77	5,40	
3	1	0,00	0,00	0,00	0,00	0,00	9,52	31,03	0,00	0,00	0,00	9,52	31,03	0,00	0,00	0,00
	2	-9,09	0,00	0,00	0,00		-9,09	0,00	0,00	0,00		0,00	0,00	0,00	0,00	
Total				0,72	-1,46	-9,15			3,23	5,20	-8,63			2,51	6,66	0,52



**Figura 26: Resumen de gráficas línea 1, servidores 1 y 2, Escenario 1**



**Figura 27: Trabajos terminados en línea 1 y 2, Escenario 1**

(—) Escenario Base (---) Ctrl Convencional con falla ( - . - ) CRC-H con falla

#### 4.4 Escenario 2: Falla en el servidor $S_{11}$ , sin recuperación

En la se resumen los parámetros asociados al escenario 2.

Tabla 17: Parámetros del escenario 2

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución (Te) y recuperación (Tr) de falla
$S_{11}$	Sin recuperación	$90 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$Te = 400$ [Unidad de Tiempo]

En la Tabla 18 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 19 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 2 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 1, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme trabajos hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en cola de 40%, 45% y 33,33% respectivamente en relación al escenario base. Lo anterior no ocurre en el control convencional en el cual la máxima cantidad de trabajos en cola del servidor  $S_{11}$  aumenta significativamente de 12 a 82 (+74,47%), mientras que el resto de las líneas se mantienen con un rendimiento idéntico respecto al escenario base.

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,92 este valor es incluso mayor respecto al escenario base que tiene una utilización máxima de 0,85, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 4,26%. Al final de la simulación la utilización del servidor sin falla es de 0,65, valor que es mayor al escenario base que tiene una utilización de 0,72, lo cual implica que el tiempo de ocio de la máquina ha aumentado en 5,32%. Respecto del control convencional, CRC-H iguala la utilización máxima del servidor  $S_{12}$  en 0,85, lo cual se traduce en que no existe una disminución en el tiempo de ocio servidor  $S_{12}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{12}$  de 0,30 a 0,65 respecto al control convencional, lo que se traduce en una disminución de 36,47% en el tiempo de ocio servidor  $S_{12}$  ver Figura

27. La baja utilización del servidor  $S_{12}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{12}$  provocada por la falla en el servidor  $S_{11}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 1,82% y aumenta la utilización final total en 5,61% respecto del control convencional bajo las condiciones del escenario 2.

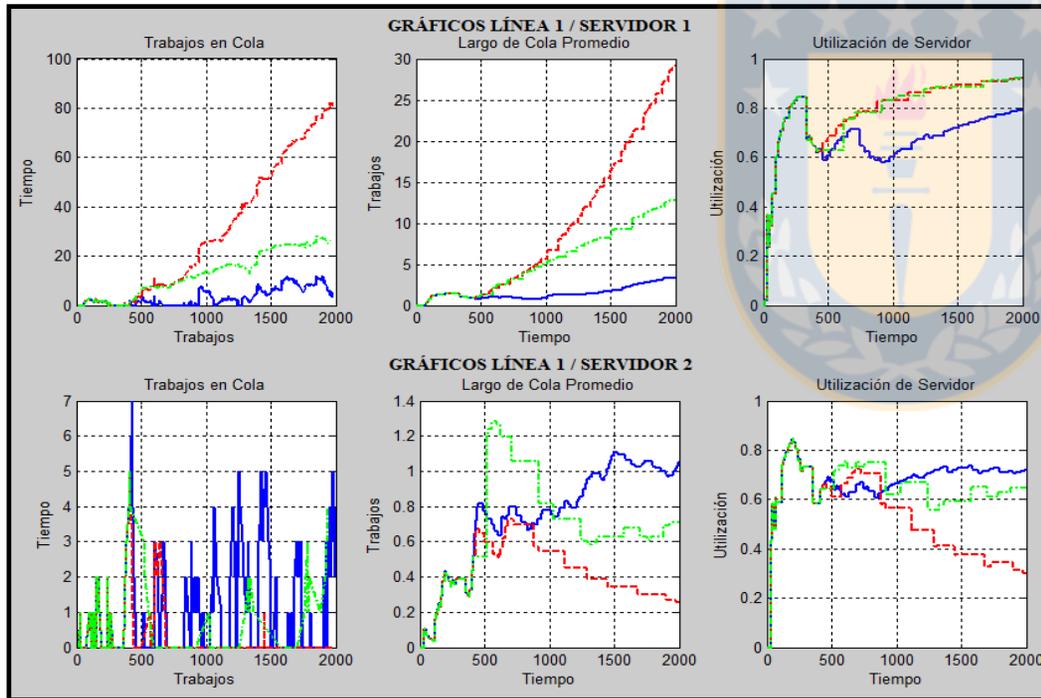
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{11}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 116 a 36 equivalente a un -52,63%, mientras que el control convencional disminuye de 116 a 40 equivalente a un -48,72%. Lo anterior indica que CRC-H produce cuatro trabajos menos que el control convencional, es decir, CRC-H disminuye en 5,26% los trabajos terminados en la línea en falla respecto al control convencional. Sin embargo, se observa una mayor producción en  $L2$  respecto al escenario base (+4,49%) y al control convencional (+4,49%), ver Figura 28, lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 270 a 277 equivalente a +1,28% respecto del control convencional bajo las condiciones del escenario 2.

**Tabla 18: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 2**

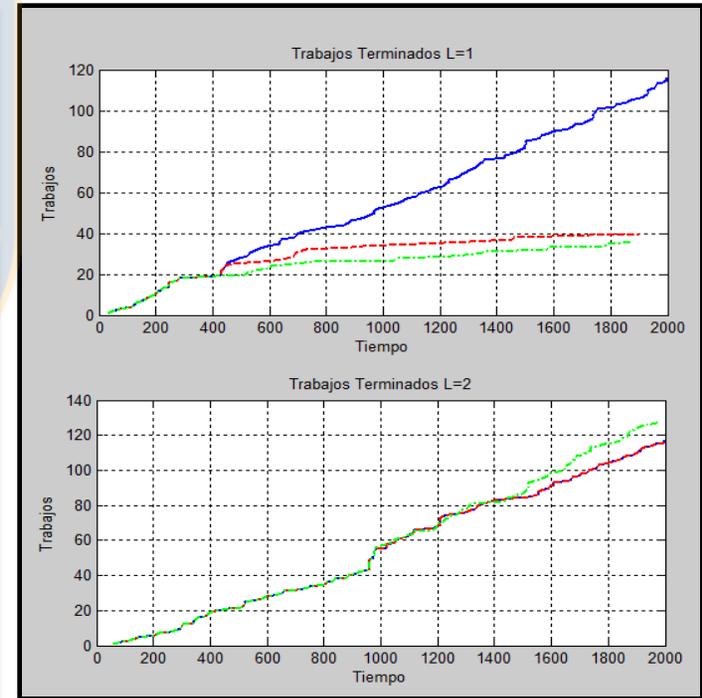
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	82	82	0,92	0,92	40	28	27	0,92	0,92	36
	2	5	0	0,85	0,30		5	4	0,85	0,65	
2	1	11	0	0,82	0,82	117	29	24	0,94	0,94	128
	2	9	3	0,83	0,82		18	9	0,92	0,92	
3	1	19	10	0,92	0,92	113	27	23	0,92	0,92	113
	2	10	5	0,91	0,91		10	5	0,91	0,91	
Total				5,26	4,70	270			5,46	5,25	277

**Tabla 19: Comparación porcentual entre controles, Escenario 2**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	74,47	90,70	4,32	7,59	-48,72	40,00	74,19	4,26	7,53	-52,63	-49,09	-50,46	-0,06	-0,06	-5,26
	2	-16,67	-100,00	0,00	-41,00		-16,67	33,33	0,00	-5,32		0,00	100,00	0,00	36,47	
2	1	0,00	0,00	0,00	0,00	0,00	45,00	100,00	6,38	6,81	4,49	45,00	100,00	6,38	6,81	4,49
	2	0,00	0,00	0,00	0,00		33,33	50,00	4,77	5,40		33,33	50,00	4,77	5,40	
3	1	0,00	0,00	0,00	0,00	0,00	17,39	39,39	0,00	0,00	0,00	17,39	39,39	0,00	0,00	0,00
	2	-9,09	0,00	0,00	0,00		-9,09	0,00	0,00	0,00		0,00	0,00	0,00	0,00	
Total				0,73	-2,98	-12,34			2,55	2,63	-11,08			1,82	5,61	1,28



**Figura 28: Resumen de gráficas línea 1, servidores 1 y 2, Escenario 2**



**Figura 29: Trabajos terminados en línea 1 y 2, Escenario 2**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

#### 4.5 Escenario 3: Falla en el servidor $S_{21}$ , sin recuperación

En la se resumen los parámetros asociados al escenario 3.

Tabla 20: Parámetros del escenario 3

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución ( $T_e$ ) y recuperación ( $T_r$ ) de falla
$S_{21}$	Sin recuperación	$45 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$T_e = 400$ [Unidad de Tiempo]

En la Tabla 21 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 22 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 3 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 2, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en cola de 27,27%, 33,33% y 2,56% respectivamente en relación al escenario base. Lo anterior no ocurre en el control convencional en el cual la máxima cantidad de trabajos en cola del servidor  $S_{21}$  aumenta significativamente de 12 a 66 (+71,43%), mientras que el resto de las líneas se mantienen con un rendimiento idéntico respecto al escenario base.

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,91 este valor es incluso mayor respecto al escenario base que tiene una utilización máxima de 0,82, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 1,94%. Al final de la simulación la utilización del servidor sin falla es de 0,91, valor que es mayor al escenario base que tiene una utilización de 0,82, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 4,76%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{22}$  de 0,65 a 0,91, lo cual se traduce en una disminución del 16,33% en el tiempo de ocio servidor  $S_{22}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{22}$  de 0,48 a 0,91 respecto al control convencional, lo que se traduce en una disminución del 30,53% en el tiempo de ocio servidor

$S_{22}$  ver Figura 29. La baja utilización del servidor  $S_{22}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{22}$  provocada por la falla en el servidor  $S_{21}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 3,44% y aumenta la utilización final total en 7% respecto del control convencional bajo las condiciones del escenario 3.

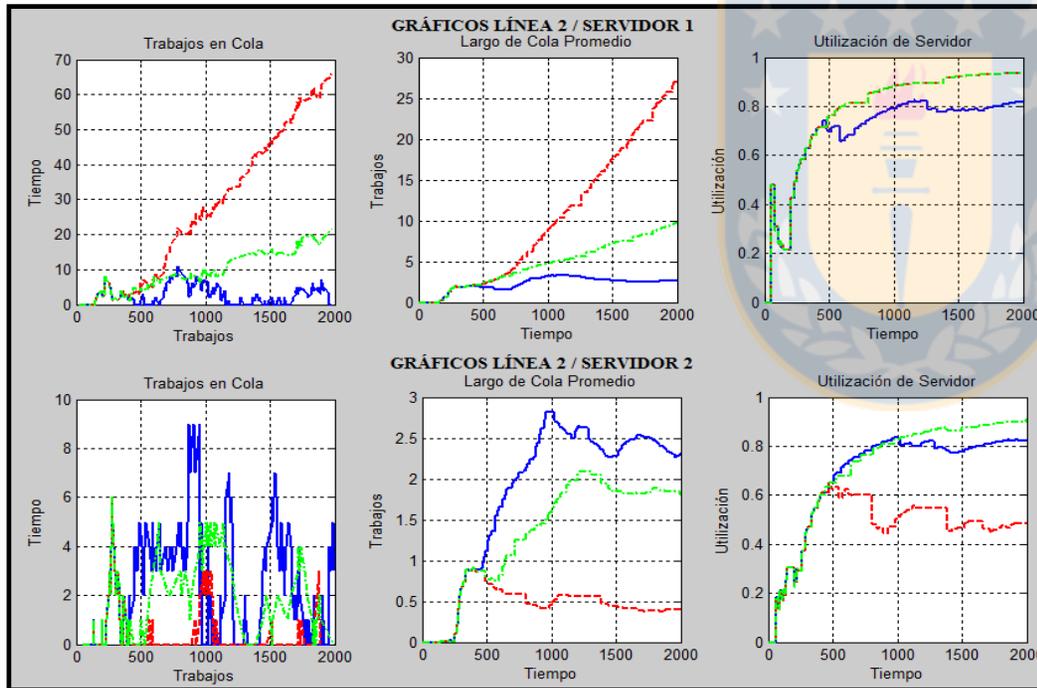
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{21}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 117 a 53 equivalente a un -37,65%, mientras que el control convencional disminuye de 117 a 58 equivalente a un -36,84%. Lo anterior indica que CRC-H produce cinco trabajos menos que el control convencional, es decir, CRC-H disminuye en 0,93% los trabajos terminados en la línea en falla respecto al control convencional. Sin embargo, se observa una mayor producción en  $L3$  respecto al escenario base (+0,44%) y al control convencional (+0,44%), ver Figura 30, lo anterior se debe a que CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H iguala la cantidad de trabajos terminados respecto del control convencional bajo las condiciones del escenario 3.

**Tabla 21: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 3**

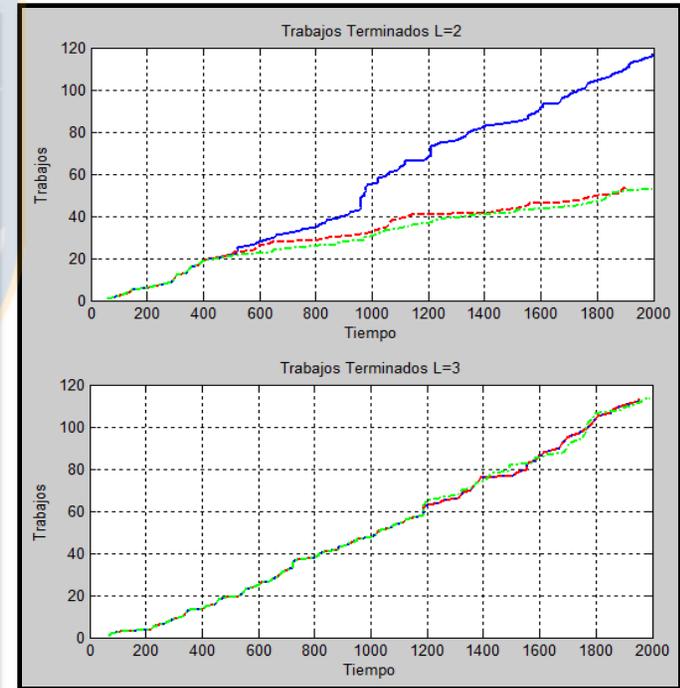
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	12	4	0,85	0,79	116	21	19	0,90	0,90	116
	2	7	2	0,85	0,72		21	15	0,88	0,88	
2	1	66	66	0,94	0,94	54	22	22	0,94	0,94	53
	2	6	0	0,65	0,48		6	0	0,91	0,91	
3	1	19	10	0,92	0,92	113	20	17	0,92	0,92	114
	2	10	5	0,91	0,91		19	13	0,93	0,93	
Total				5,11	4,76	283			5,48	5,48	283

**Tabla 22: Comparación porcentual entre controles, Escenario 3**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					CRC-H vs Control Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	0,00	0,00	0,00	0,00	0,00	27,27	65,22	3,20	6,47	0,00	27,27	65,22	3,20	6,47	0,00
	2	0,00	0,00	0,00	0,00	0,00	50,00	76,47	1,94	10,16	0,00	50,00	76,47	1,94	10,16	0,00
2	1	71,43	100,00	6,39	6,82	-36,84	33,33	100,00	6,39	6,82	-37,65	-50,00	-50,00	0,00	0,00	-0,93
	2	-20,00	-100,00	-12,29	-26,14	-36,84	-20,00	-100,00	4,13	4,76	-37,65	0,00	0,00	16,33	30,53	-0,93
3	1	0,00	0,00	0,00	0,00	0,00	2,56	25,93	-0,17	-0,17	0,44	2,56	25,93	-0,17	-0,17	0,44
	2	-9,09	0,00	0,00	0,00	0,00	22,58	44,44	1,26	1,26	0,44	31,03	44,44	1,26	1,26	0,44
Total				-0,68	-2,28	-10,02			2,76	4,73	-10,02			3,44	7,00	0,00



**Figura 30: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 3**



**Figura 31: Trabajos terminados en línea 2 y 3, Escenario 3**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

#### 4.6 Escenario 4: Falla en el servidor $S_{21}$ , sin recuperación

En la se resumen los parámetros asociados al escenario 4.

Tabla 23: Parámetros del escenario 4

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución ( $T_e$ ) y recuperación ( $T_r$ ) de falla
$S_{21}$	Sin recuperación	$90 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$T_e = 400$ [Unidad de Tiempo]

En la Tabla 24 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 25 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 4 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 2, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en cola de 40%, 46,34% y 17,39% respectivamente en relación al escenario base. Lo anterior no ocurre en el control convencional en el cual la máxima cantidad de trabajos en cola del servidor  $S_{21}$  aumenta significativamente de 11 a 80 (+75,82%), mientras que el resto de las líneas se mantienen con un rendimiento idéntico respecto al escenario base.

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,86 este valor es incluso mayor respecto al escenario base que tiene una utilización máxima de 0,83, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 1,49%. Al finalizar la simulación la utilización del servidor sin falla es de 0,86, valor que es mayor al escenario base que tiene una utilización de 0,82, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 2,13%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{22}$  de 0,61 a 0,86, lo cual se traduce en una disminución del 16,92% en el tiempo de ocio servidor  $S_{22}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{22}$  de 0,28 a 0,86 respecto al control convencional, lo que se traduce en una disminución del 51,28% en el tiempo de ocio servidor

$S_{22}$ , ver Figura 31. La baja utilización del servidor  $S_{22}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{22}$  provocada por la falla en el servidor  $S_{21}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 3,23% y aumenta la utilización final total en 8,59% respecto del control convencional bajo las condiciones del escenario 4.

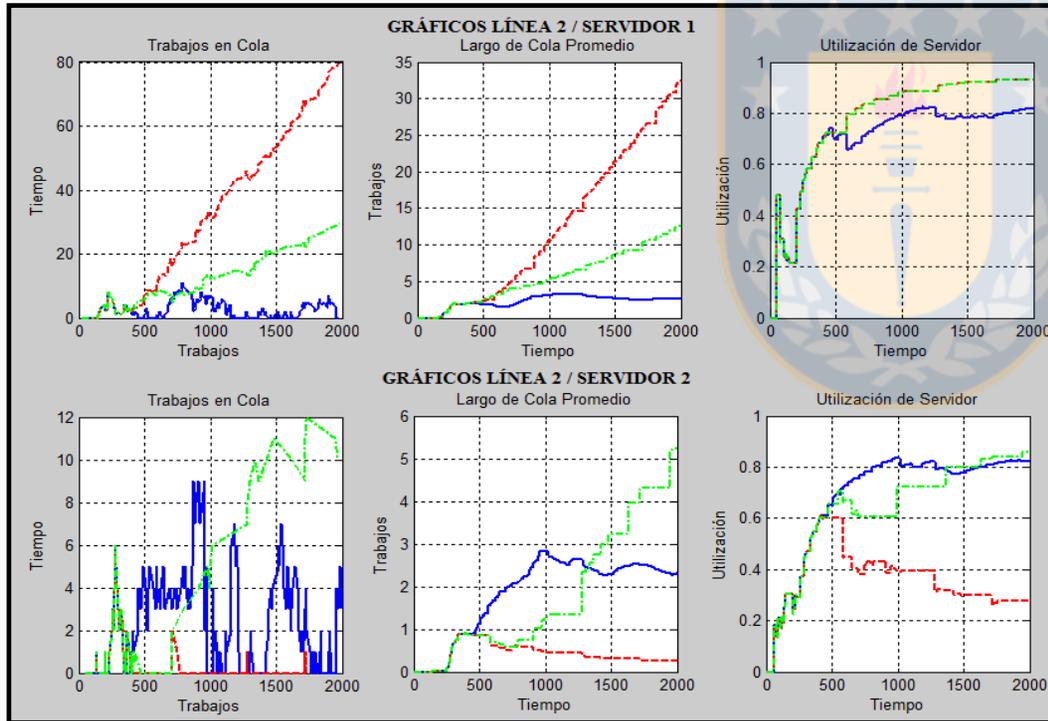
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{21}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional ver Figura 32. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 117 a 29 equivalente a un -60,27%, mientras que el control convencional disminuye de 117 a 40 equivalente a un -49,04%. Lo anterior indica que CRC-H produce once trabajos menos que el control convencional, es decir, CRC-H disminuye en 15,94% los trabajos terminados en la línea en falla respecto al control convencional. En términos globales CRC-H disminuye el total de trabajos terminados de 269 a 258 equivalente a -2,09% respecto del control convencional bajo las condiciones del escenario 4.

**Tabla 24: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 4**

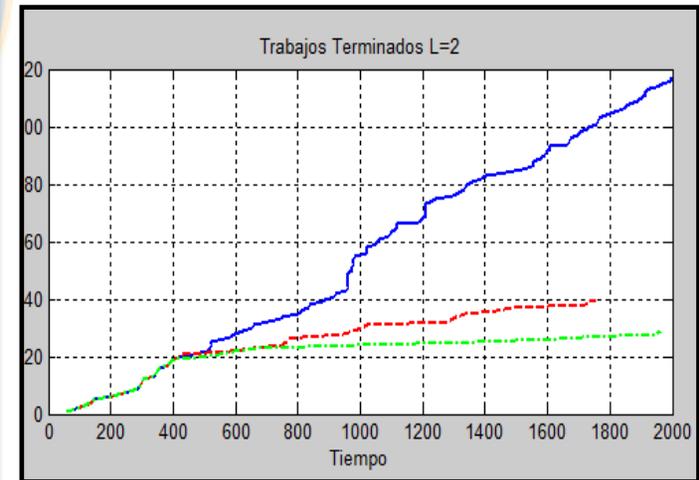
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	12	4	0,85	0,79	116	28	26	0,90	0,90	116
	2	7	2	0,85	0,72		22	16	0,88	0,88	
2	1	80	80	0,93	0,93	40	30	30	0,93	0,93	29
	2	6	0	0,61	0,28		12	10	0,86	0,86	
3	1	19	10	0,92	0,92	113	27	24	0,92	0,92	113
	2	10	5	0,91	0,91		10	5	0,91	0,91	
Total				5,07	4,55	269			5,41	5,41	258

**Tabla 25: Comparación porcentual entre controles, Escenario 4**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	0,00	0,00	0,00	0,00	0,00	40,00	73,33	3,22	6,49	0,00	40,00	73,33	3,22	6,49	0,00
	2	0,00	0,00	0,00	0,00	0,00	51,72	77,78	1,94	10,16	0,00	51,72	77,78	1,94	10,16	0,00
2	1	75,82	100,00	6,10	6,52	-49,04	46,34	100,00	6,10	6,52	-60,27	-45,45	-45,45	0,00	0,00	-15,94
	2	-20,00	-100,00	-15,47	-49,70		14,29	53,85	1,49	2,13		33,33	100,00	16,92	51,28	
3	1	0,00	0,00	0,00	0,00	0,00	17,39	41,18	0,00	0,00	0,00	17,39	41,18	0,00	0,00	0,00
	2	-9,09	0,00	0,00	0,00	0,00	-9,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Total				-1,14	-4,54	-12,52			2,10	4,07	-14,57			3,23	8,59	-2,09



**Figura 32: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 4**



**Figura 33: Trabajos terminados en línea 2, Escenario 4**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

#### 4.7 Escenario 5: Falla en el servidor $S_{31}$ , sin recuperación

En la se resumen los parámetros asociados al escenario 5.

Tabla 26: Parámetros del escenario 5

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución (Te) y recuperación (Tr) de falla
$S_{31}$	Sin recuperación	$45 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$Te = 400$ [Unidad de Tiempo]

En la Tabla 27 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 28 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 5 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 3, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en cola de 20%, 24,14% y 0% respectivamente en relación al escenario base. Lo anterior no ocurre en el control convencional en el cual la máxima cantidad de trabajos en cola del servidor  $S_{31}$  aumenta significativamente de 19 a 81 (+62%), mientras que el resto de las líneas se mantienen con un rendimiento idéntico respecto al escenario base.

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,75 este valor es menor respecto al escenario base que tiene una utilización máxima de 0,91, lo cual implica que el tiempo de ocio de la máquina ha aumentado en un 10%. Al final de la simulación la utilización del servidor sin falla es de 0,68, valor que es menor al escenario base que tiene una utilización de 0,91, lo cual implica que el tiempo de ocio de la máquina se ha aumentado en un -14,44%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{32}$  de 0,74 a 0,75, lo cual se traduce en una disminución del 0,35% en el tiempo de ocio servidor  $S_{32}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{32}$  de 0,35 a 0,68 respecto al control convencional, lo que se traduce en una disminución del 32,67% en el tiempo de ocio servidor  $S_{32}$ , ver Figura 33. La

baja utilización del servidor  $S_{32}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{32}$  provocada por la falla en el servidor  $S_{31}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 2,77% y aumenta la utilización final total en 8,46% respecto del control convencional bajo las condiciones del escenario 5.

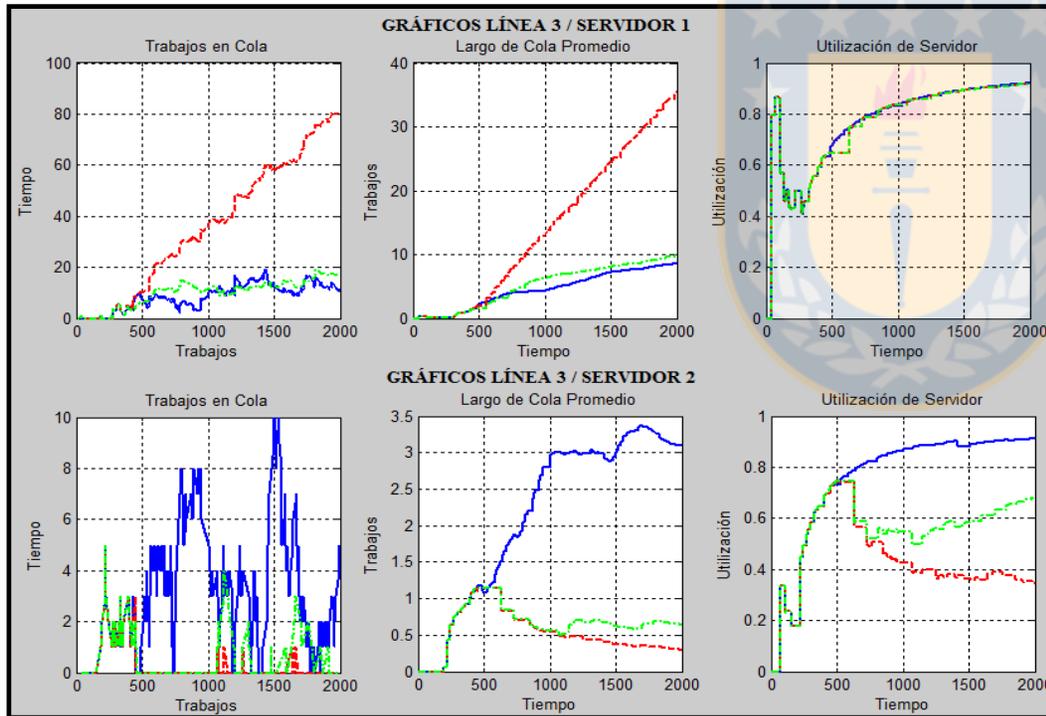
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{31}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 113 a 49 equivalente a un -39,51%, mientras que el control convencional de igual forma disminuye la cantidad de trabajos terminados de 113 a 49. Sin embargo, se observa una mayor producción en  $L2$  respecto al escenario base (+4,49%) y al control convencional (+4,49%), ver Figura 34, lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 282 a 293 equivalente a +1,91% respecto del control convencional bajo las condiciones del escenario 5.

**Tabla 27: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 5**

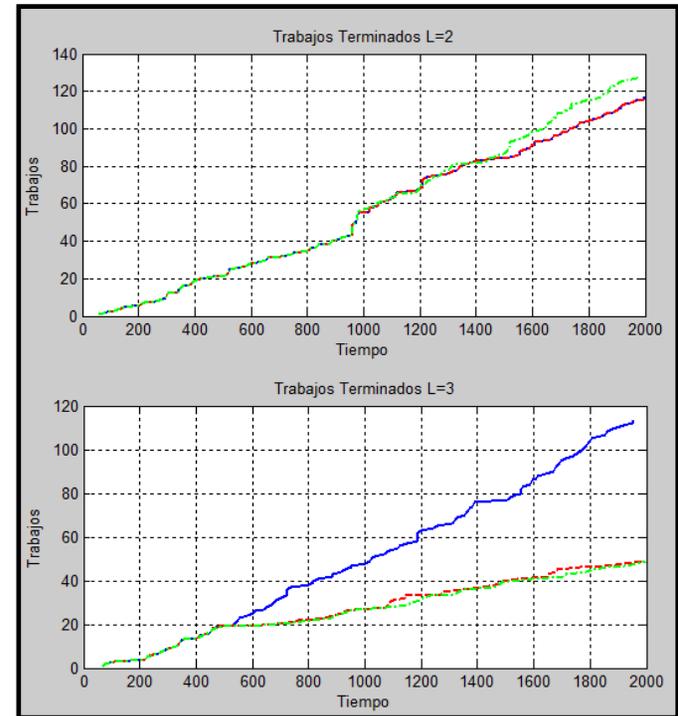
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	12	4	0,85	0,79	116	18	16	0,90	0,90	116
	2	7	2	0,85	0,72	0	18	12	0,88	0,88	0
2	1	11	0	0,82	0,82	117	18	17	0,93	0,93	128
	2	9	3	0,83	0,82	0	21	15	0,92	0,92	0
3	1	81	80	0,92	0,92	49	19	17	0,92	0,92	49
	2	5	0	0,74	0,35	0	5	0	0,75	0,68	0
Total				5,01	4,42	282			5,30	5,23	293

**Tabla 28: Comparación porcentual entre controles, Escenario 5**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	0,00	0,00	0,00	0,00	0,00	20,00	60,00	3,05	6,32	0,00	20,00	60,00	3,05	6,32	0,00
	2	0,00	0,00	0,00	0,00	0,00	44,00	71,43	1,94	10,16	0,00	44,00	71,43	1,94	10,16	0,00
2	1	0,00	0,00	0,00	0,00	0,00	24,14	100,00	6,28	6,70	4,49	24,14	100,00	6,28	6,70	4,49
	2	0,00	0,00	0,00	0,00	0,00	40,00	66,67	4,77	5,40	4,49	40,00	66,67	4,77	5,40	4,49
3	1	62,00	77,78	-0,02	-0,02	-39,51	0,00	25,93	-0,02	-0,02	-39,51	-62,00	-64,95	0,00	0,00	0,00
	2	-41,18	-100,00	-10,35	-44,98	-39,51	-41,18	-100,00	-10,00	-14,44	-39,51	0,00	0,00	0,35	32,67	0,00
Total				-1,68	-6,02	-10,19			1,09	2,45	-8,29			2,77	8,46	1,91



**Figura 34: Resumen de gráficas línea 3, servidores 1 y 2, Escenario 5**



**Figura 35: Trabajos terminados en línea 2 y 3, Escenario 5**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

#### 4.8 Escenario 6: Falla en el servidor $S_{31}$ , sin recuperación

En la se resumen los parámetros asociados al escenario 6.

Tabla 29: Parámetros del escenario 6

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución (Te) y recuperación (Tr) de falla
$S_{31}$	Sin recuperación	$90 \left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$Te = 400$ [Unidad de Tiempo]

En la Tabla 30 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 31 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 6 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 3, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en cola de 29,41%, 38,89% y 24% respectivamente en relación al escenario base. Lo anterior no ocurre en el control convencional en el cual la máxima cantidad de trabajos en cola del servidor  $S_{31}$  aumenta significativamente de 19 a 98 (+67,52%), mientras que el resto de las líneas se mantienen con un rendimiento idéntico respecto al escenario base.

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,9 este valor es menor respecto al escenario base que tiene una utilización máxima de 0,91, lo cual implica que el tiempo de ocio de la máquina se ha aumentado en 0,72%. Al final de la simulación la utilización del servidor sin falla es de 0,82, valor que es menor al escenario base que tiene una utilización de 0,91, lo cual implica que el tiempo de ocio de la máquina ha aumentado en 5,48%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{32}$  de 0,73 a 0,9, lo cual se traduce en una disminución del 10,62% en el tiempo de ocio servidor  $S_{32}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{32}$  de 0,24 a 0,82 respecto al control convencional, lo que se traduce en una disminución del 55,26% en el tiempo de ocio servidor  $S_{32}$ , ver Figura 35. La

baja utilización del servidor  $S_{32}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{32}$  provocada por la falla en el servidor  $S_{31}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 4,27% y aumenta la utilización final total en 10,76% respecto del control convencional bajo las condiciones del escenario 6.

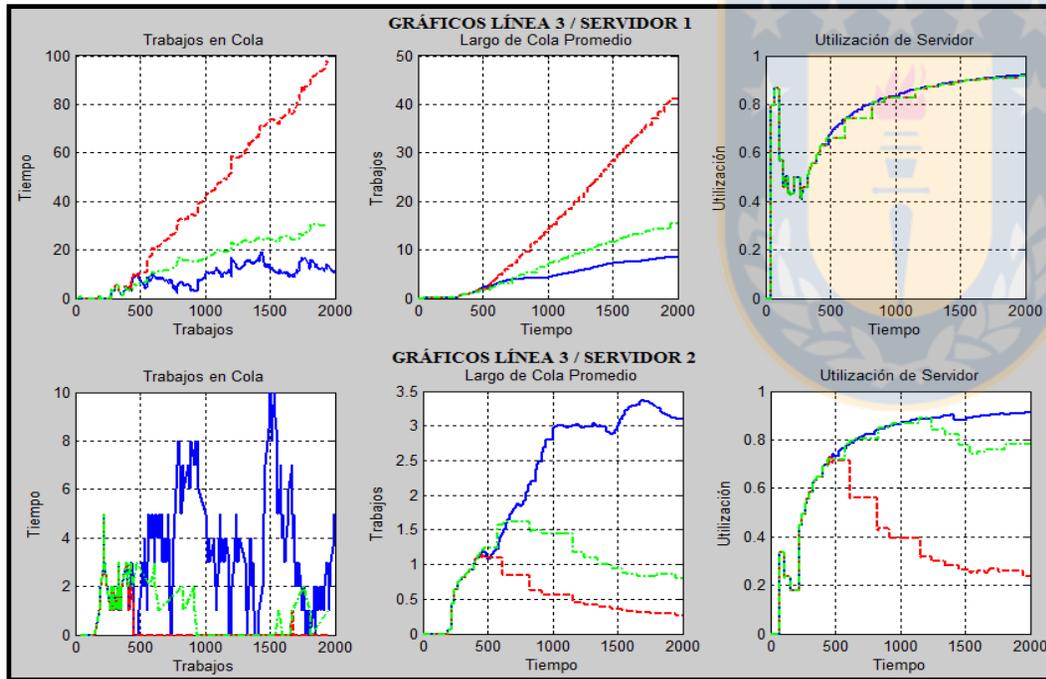
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{31}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional, ver Figura 36. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 113 a 50 equivalente a un -56,94%, mientras que el control convencional disminuye de 113 a 32 equivalente a un -55,86%. Lo anterior indica que CRC-H produce un trabajo menos que el control convencional, es decir, CRC-H disminuye en 1,59% los trabajos terminados en la línea en falla respecto al control convencional. Sin embargo, se observa una mayor producción en  $L2$  respecto al escenario base (+6,67%) y al control convencional (+6,67%), lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 265 a 279 equivalente a +2,57% respecto del control convencional bajo las condiciones del escenario 6.

**Tabla 30: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 6**

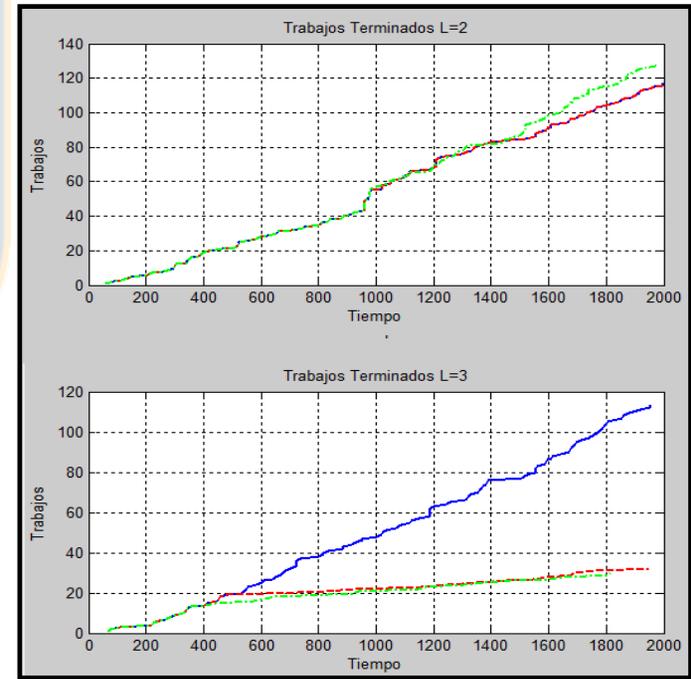
Línea	Servidor	Control Convencional c/Falla				Trabajos Terminados	Control CRC-H				Trabajos Terminados
		Trabajos en cola		Utilización			Trabajos en cola		Utilización		
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	12	4	0,85	0,79	116	22	20	0,90	0,90	114
	2	7	2	0,85	0,72		21	15	0,88	0,88	
2	1	11	0	0,82	0,82	117	25	20	0,96	0,96	134
	2	9	3	0,83	0,82		21	4	0,88	0,87	
3	1	98	97	0,92	0,92	32	31	30	0,92	0,92	31
	2	5	0	0,73	0,24		6	1	0,90	0,82	
Total				5,00	4,31	265			5,44	5,34	279

**Tabla 31: Comparación porcentual entre controles, Escenario 6**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	0,00	0,00	0,00	0,00	0,00	29,41	66,67	3,19	6,46	-0,87	29,41	66,67	3,19	6,46	-0,87
	2	0,00	0,00	0,00	0,00	0,00	50,00	76,47	1,82	10,05	-0,87	50,00	76,47	1,82	10,05	-0,87
2	1	0,00	0,00	0,00	0,00	0,00	38,89	100,00	7,60	8,03	6,77	38,89	100,00	7,60	8,03	6,77
	2	0,00	0,00	0,00	0,00	0,00	40,00	14,29	2,87	2,57	6,77	40,00	14,29	2,87	2,57	6,77
3	1	67,52	81,31	-0,10	-0,10	-55,86	24,00	50,00	-0,06	-0,06	-56,94	-51,94	-52,76	0,05	0,05	-1,59
	2	-41,18	-100,00	-11,32	-58,95	-55,86	-33,33	-66,67	-0,72	-5,48	-56,94	9,09	100,00	10,62	55,26	-1,59
Total				-1,84	-7,30	-13,26			2,43	3,49	-10,72			4,27	10,76	2,57



**Figura 36: Resumen de gráficas línea 3, servidores 1 y 2, Escenario 6**



**Figura 37: Trabajos terminados en línea 2 y 3, Escenario 6**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

#### 4.9 Escenario 7: Falla en el servidor $S_{II}$ , con recuperación

En la se resumen los parámetros asociados al escenario 7.

Tabla 32: Parámetros del escenario 7

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución ( $T_e$ ) y recuperación ( $T_r$ ) de falla
$S_{II}$	Con recuperación	45 $\left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	$T_e = 400$ [Unidad de Tiempo] $T_r = 1200$ [Unidad de Tiempo]

En la Tabla 33 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 34 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 7 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 1, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en colas  $L1$  y  $L2$  de 25%, 15,38% respectivamente mientras que se disminuyen los trabajos en la cola de  $L3$  en 8,57 en relación al escenario base. Se observa que a pesar de la distribución inteligente de trabajos existe una mayor cantidad de trabajos en la cola de la línea en falla. En el control convencional la máxima cantidad de trabajos en cola del servidor  $S_{II}$  aumenta significativamente de 12 a 27 (+38,46%) y en menor proporción en  $L2$  (0%) y  $L3$  (17,39%).

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,89 este valor es incluso mayor respecto al escenario base que tiene una utilización máxima de 0,85, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 2,56%. Al final de la simulación la utilización del servidor sin falla es de 0,89, valor que es mayor al escenario base que tiene una utilización de 0,72, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 10,46%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{I2}$  de 0,85 a 0,89, lo cual se traduce en una disminución del 2,56% en el tiempo de ocio servidor  $S_{I2}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{I2}$  de 0,67 a 0,89 respecto al control

convencional, ver Figura 37, lo que se traduce en una disminución del 14,34% en el tiempo de ocio servidor  $S_{12}$ . La baja utilización del servidor  $S_{12}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{12}$  provocada por la falla en el servidor  $S_{11}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 0,97% y aumenta la utilización final total en 3,07% respecto del control convencional bajo las condiciones del escenario 7.

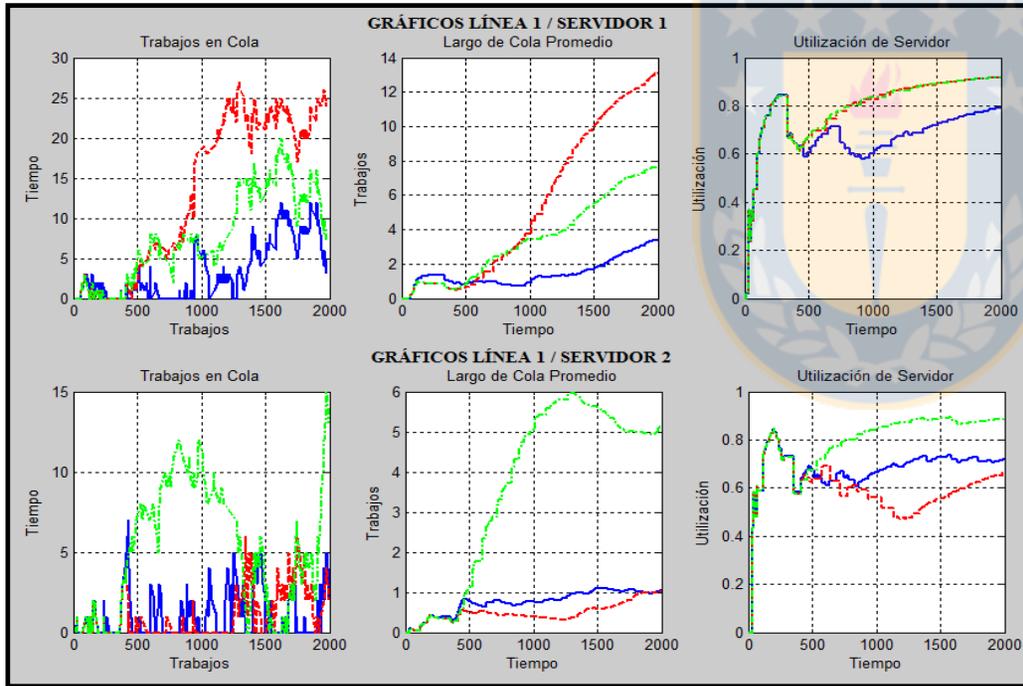
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{11}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional, ver Figura 38. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 116 a 83 equivalente a un -16,58% mientras que el control convencional disminuye de 116 a 87 equivalente a un -14,29%. Lo anterior indica que CRC-H produce cuatro trabajos menos que el control convencional, es decir, CRC-H disminuye en 2,35% los trabajos terminados en la línea en falla respecto al control convencional. Sin embargo, se observa una mayor producción en  $L2$  respecto al escenario base (+6,77%) y al control convencional (+7,63%), lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 317 a 332 equivalente a +2,31% respecto del control convencional bajo las condiciones del escenario 7.

**Tabla 33: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 7**

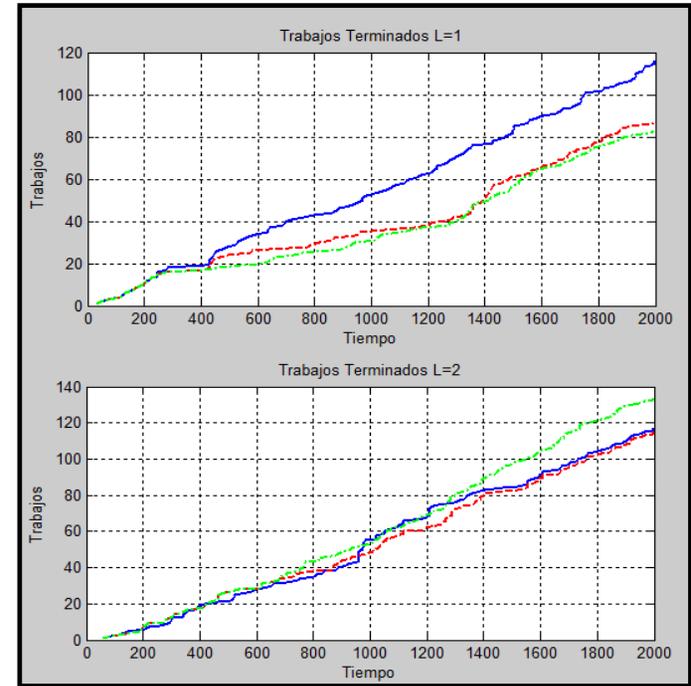
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	27	25	0,92	0,92	87	20	8	0,92	0,92	83
	2	6	2	0,85	0,67		15	13	0,89	0,89	
2	1	11	0	0,92	0,87	115	15	0	0,96	0,96	134
	2	12	1	0,86	0,85		21	3	0,88	0,87	
3	1	27	21	0,92	0,92	115	16	10	0,92	0,92	115
	2	12	4	0,93	0,93		12	4	0,93	0,93	
Total				5,40	5,15	317			5,50	5,48	332

**Tabla 34: Comparación porcentual entre controles, Escenario 7**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	38,46	72,41	4,14	7,41	-14,29	25,00	33,33	4,17	7,44	-16,58	-14,89	-51,52	0,03	0,03	-2,35
	2	-7,69	0,00	0,00	-3,94		36,36	73,33	2,56	10,46		42,86	73,33	2,56	14,34	
2	1	0,00	0,00	5,74	3,07	-0,86	15,38	0,00	7,59	7,91	6,77	15,38	0,00	1,86	4,85	7,63
	2	14,29	-50,00	1,37	1,75		40,00	0,00	2,87	2,57		27,27	50,00	1,50	0,83	
3	1	17,39	35,48	0,05	0,05	0,88	-8,57	0,00	0,05	0,05	0,88	-25,58	-35,48	0,00	0,00	0,00
	2	0,00	-11,11	0,89	0,89		0,00	-11,11	0,89	0,89		0,00	0,00	0,00	0,00	
Total				2,02	1,68	-4,37			2,99	4,75	-2,06			0,97	3,07	2,31



**Figura 38: Resumen de gráficas línea 3, servidores 1 y 2, Escenario 7**



**Figura 39: Trabajos terminados en línea 2 y 3, Escenario 7**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

#### 4.10 Escenario 8: Falla en el servidor $S_{21}$ , con recuperación

En la se resumen los parámetros asociados al escenario 8.

Tabla 35: Parámetros del escenario 8

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución (Te) y recuperación (Tr) de falla
$S_{21}$	Con recuperación	45 $\left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	Te = 400 [Unidad de Tiempo] Tr = 1200 [Unidad de Tiempo]

En la se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 37 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 8 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 2, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en colas de 7,69%, 18,52 y 9,52% respectivamente en relación al escenario base. Se observa que a pesar de la distribución inteligente de trabajos existe una mayor cantidad de trabajos en la cola de la línea en falla. En el control convencional la máxima cantidad de trabajos en cola del servidor  $S_{21}$  aumenta significativamente de 11 a 37 (+54,17%) y en menor proporción en  $L1$  (-4,35%) y  $L3$  (+17,39%).

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,91 este valor es incluso mayor respecto al escenario base que tiene una utilización máxima de 0,83, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 4,27%. Al final de la simulación la utilización del servidor sin falla es de 0,91, valor que es mayor al escenario base que tiene una utilización de 0,82, lo cual implica que el tiempo de ocio de la máquina se ha reducido en un 4,81%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{22}$  de 0,75 a 0,91, lo cual se traduce en una disminución del 9,8% en el tiempo de ocio servidor  $S_{22}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{22}$  de 0,59 a 0,91 respecto al control

convencional, lo que se traduce en una disminución del 21,07% en el tiempo de ocio servidor  $S_{22}$ , ver Figura 39. La baja utilización del servidor  $S_{22}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{22}$  provocada por la falla en el servidor  $S_{11}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 2,23% y aumenta la utilización final total en 5,73% respecto del control convencional bajo las condiciones del escenario 8.

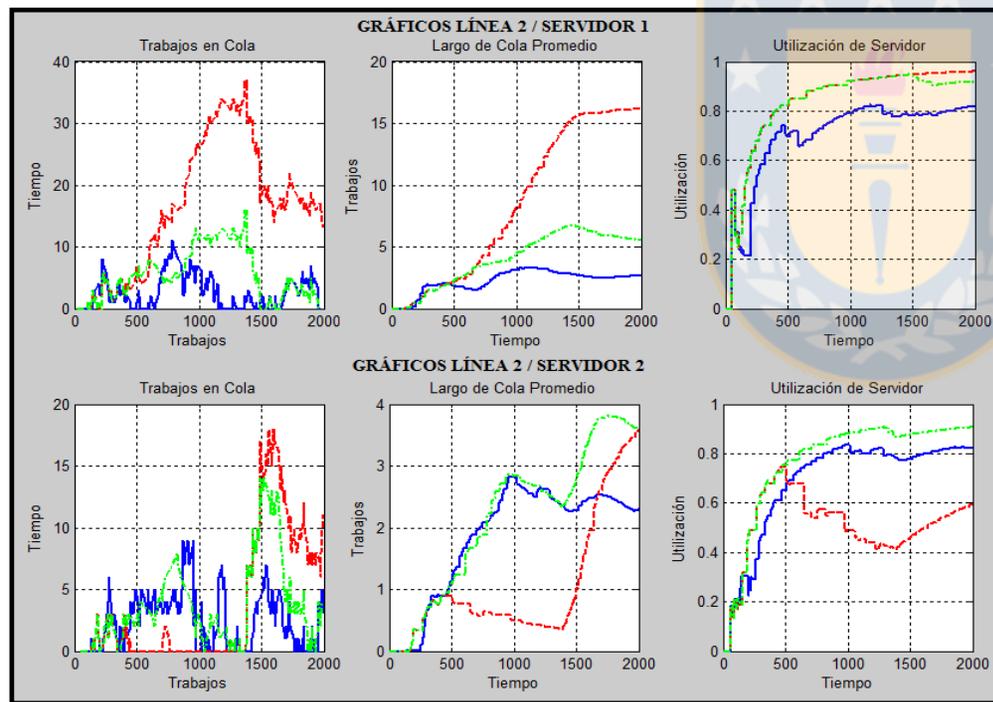
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{21}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional, ver Figura 40. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 117 a 91 equivalente a un -39,76%, al igual que en el control convencional. Sin embargo, se observa una mayor producción en  $L1$  respecto al control convencional (+1,79%), lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a esta línea aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 316 a 320 equivalente a +0,63% respecto del control convencional bajo las condiciones del escenario 8.

**Tabla 36: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 8**

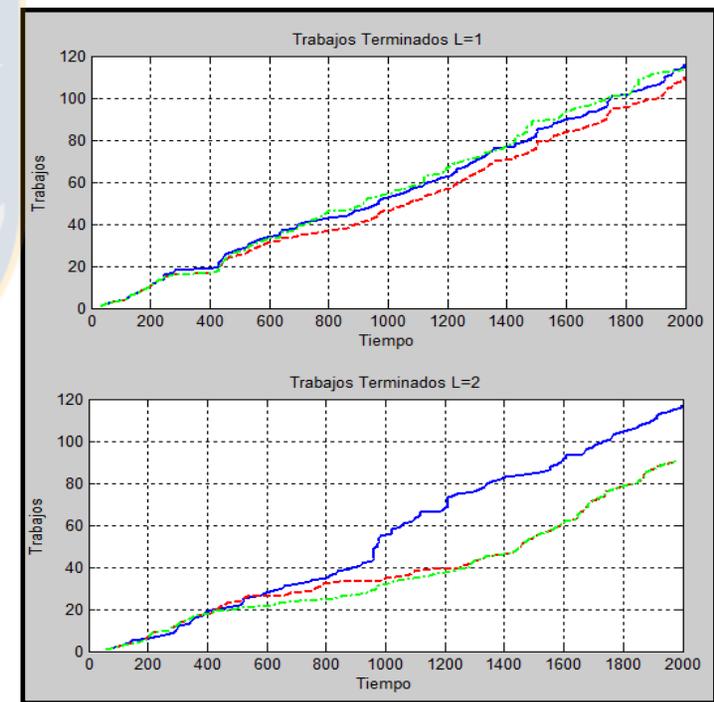
Línea	Servidor	Control Convencional c/Falla					Control CRC-H				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	11	2	0,84	0,78	110	14	10	0,90	0,90	114
	2	6	2	0,85	0,68	0	21	15	0,88	0,88	0
2	1	37	13	0,96	0,96	91	16	0	0,95	0,92	91
	2	18	11	0,75	0,59	0	14	4	0,91	0,91	0
3	1	27	21	0,92	0,92	115	23	17	0,92	0,92	115
	2	12	4	0,93	0,93	0	12	4	0,93	0,93	0
Total				5,25	4,86	316			5,49	5,46	320

**Tabla 37: Comparación porcentual entre controles, Escenario 8**

Línea	Servidor	Ctrl Convencional c/Falla vs Escenario Base					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final			
1	1	-4,35	-33,33	-0,19	-0,64	-2,65	7,69	42,86	3,19	6,46	-0,87	12,00	66,67	3,38	7,10	1,79
	2	-7,69	0,00	0,00	-2,68		50,00	76,47	1,82	10,05		55,56	76,47	1,82	12,70	
2	1	54,17	100,00	7,62	8,05	-12,50	18,52	0,00	6,92	5,80	-12,50	-39,62	-100,00	-0,71	-2,26	0,00
	2	33,33	57,14	-5,55	-16,43		21,74	14,29	4,27	4,81		-12,50	-46,67	9,80	21,07	
3	1	17,39	35,48	0,05	0,05	0,88	9,52	25,93	0,05	0,05	0,88	-8,00	-10,53	0,00	0,00	0,00
	2	0,00	-11,11	0,89	0,89		0,00	-11,11	0,89	0,89		0,00	0,00	0,00	0,00	0,00
Total				0,60	-1,22	-4,53			2,83	4,51	-3,90			2,23	5,73	0,63



**Figura 40: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 8**



**Figura 41: Trabajos terminados en línea 2 y 3, Escenario 8**

(—) Escenario Base (---) Ctrl Convencional con falla (···) CRC-H con falla

#### 4.11 Escenario 9: Falla en el servidor $S_{31}$ , con recuperación

En la se resumen los parámetros asociados al escenario 9.

Tabla 38: Parámetros del escenario 9

Servidor en Falla	Tipo de Falla	Valor de la media de tasa en Falla	Tiempo de ejecución (Te) y recuperación (Tr) de falla
$S_{31}$	Con recuperación	45 $\left[ \frac{\text{Trabajos}}{\text{Unidad de Tiempo}} \right]$	Te = 400 [Unidad de Tiempo] Tr = 1200 [Unidad de Tiempo]

En la Tabla 39 se resumen los resultados de las métricas asociadas a la simulación de la red controlada por un sistema convencional y la simulación con la red controlada por CRC-H, mientras que en la Tabla 40 se realiza un cuadro comparativo porcentual entre el Escenario Base (Escenario Base) y el Escenario 9 (Control Convencional y CRC-H con falla sin recuperación). A continuación se comentan las comparaciones para los distintos controles evaluados:

**Carga de trabajo en colas de Servidores.** Se puede apreciar que a pesar de que existe una falla en la línea 3, el ruteo inteligente del control CRC-H distribuye los trabajos a cada línea de manera uniforme hacia  $L1$ ,  $L2$  y  $L3$ , esto es equivalente a un aumento de trabajos en colas de 7,69%, 4,35% y 19,15% respectivamente en relación al escenario base. Se observa que a pesar de la distribución inteligente de trabajos existe una mayor cantidad de trabajos en la cola de la línea en falla. En el control convencional la máxima cantidad de trabajos en cola del servidor  $S_{31}$  aumenta significativamente de 19 a 65 (+54,76%) y en menor proporción en  $L1$  (-4,35%) y  $L2$  (0%).

**Utilización de Servidores.** Se observa que en CRC-H la utilización máxima del servidor sin falla asciende a 0,79 este valor es menor respecto al escenario base que tiene una utilización máxima de 0,91, lo cual implica que el tiempo de ocio de la máquina ha aumentado en 6,87%. Al final de la simulación la utilización del servidor sin falla es de 0,69, valor que es mayor al escenario base que tiene una utilización de 0,91, lo cual implica que el tiempo de ocio de la máquina ha aumentado en 13,91%. Respecto del control convencional, CRC-H aumenta la utilización máxima del servidor  $S_{32}$  de 0,74 a 0,79, lo cual se traduce en una disminución del 3,21% en el tiempo de ocio servidor  $S_{32}$ . Al final de la simulación, CRC-H aumenta la utilización del servidor  $S_{32}$  de 0,52 a 0,69 respecto al control convencional, lo que

se traduce en una disminución del 13,54% en el tiempo de ocio servidor  $S_{32}$ , ver Figura 41. La baja utilización del servidor  $S_{32}$  reflejada cuando se utiliza el control convencional se produce por la disminución de los trabajos enviados al servidor  $S_{32}$  provocada por la falla en el servidor  $S_{31}$ , por otro lado, la alta utilización del servidor cuando se utiliza CRC-H se produce por el control de tasa realizado localmente en la línea que presenta falla; el servidor que no está en falla disminuye su tasa a la tasa del servidor en falla manteniendo el equilibrio en la línea. En términos globales CRC-H aumenta la utilización máxima total en 1,45% y aumenta la utilización final total en 5% respecto del control convencional bajo las condiciones del escenario 9.

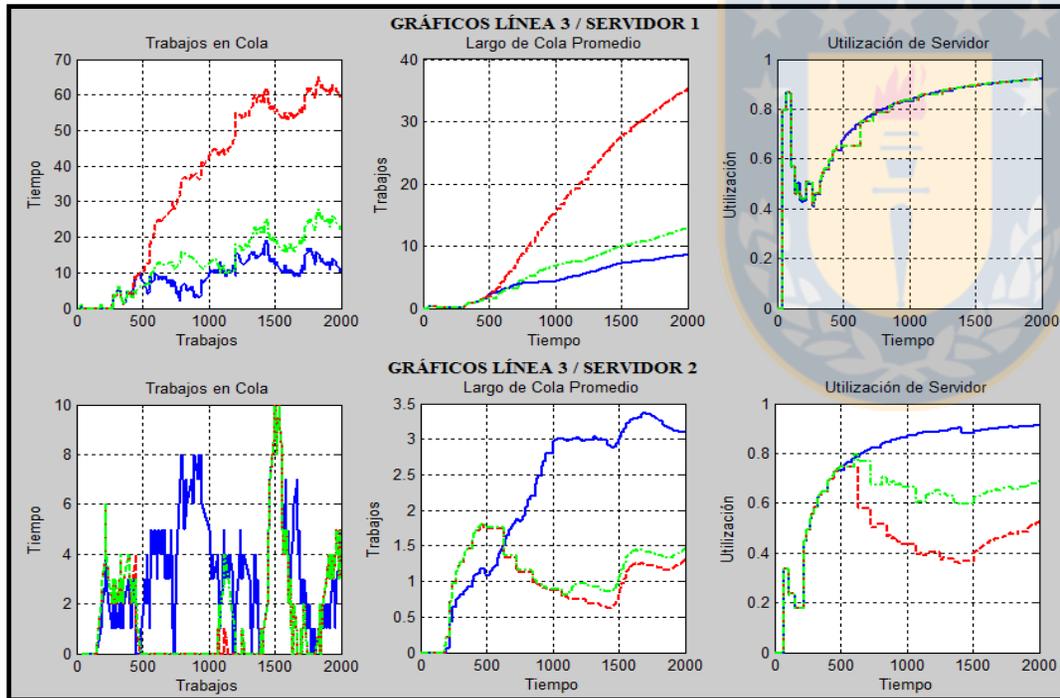
**Trabajos Terminados.** Dado que existe una falla en el servidor  $S_{31}$ , inevitablemente los trabajos terminados en la línea en falla se verán disminuidos tanto en CRC-H como en el control convencional, ver Figura 42. CRC-H respecto al escenario base disminuye los trabajos terminados en la línea en falla de 113 a 77 equivalente a un -18,95%, al igual que en el control convencional. Sin embargo, se observa una mayor producción en  $L1$  (1,35%) y  $L2$  (7,63) respecto al control convencional, lo anterior se debe a CRC-H al distribuir los trabajos de forma inteligente se disminuyen los tiempos en que las colas de las líneas están vacías, por lo tanto con una producción continua, esto se confirma con el hecho de que los servidores asociados a estas líneas aumentan su utilización. En términos globales CRC-H aumenta el total de trabajos terminados de 302 a 324 equivalente a +3,51% respecto del control convencional bajo las condiciones del escenario 9.

**Tabla 39: Resultados Métricas Control Convencional c/Falla y Control CRC-H, Escenario 9**

Línea	Servidor	Control Convencional c/Falla				Trabajos Terminados	Control CRC-H				
		Trabajos en cola		Utilización			Trabajos en cola		Utilización		
		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	11	2	0,84	0,78	110	14	8	0,87	0,84	113
	2	6	2	0,85	0,68		21	11	0,88	0,88	
2	1	11	0	0,92	0,87	115	12	0	0,94	0,91	134
	2	12	1	0,86	0,85		16	1	0,88	0,88	
3	1	65	59	0,92	0,92	77	28	22	0,92	0,92	77
	2	10	4	0,74	0,52		10	4	0,79	0,69	
Total				5,14	4,63	302			5,29	5,12	324

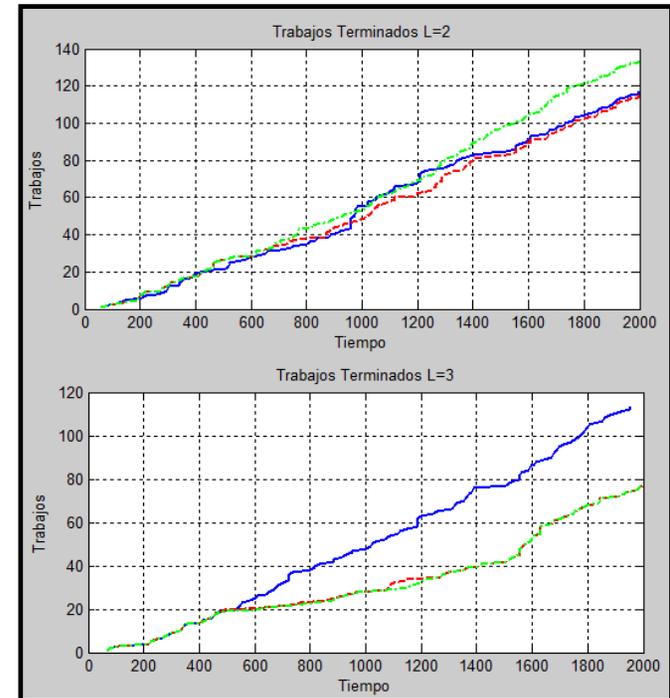
**Tabla 40: Comparación porcentual entre controles, Escenario 9**

Línea	Servidor	Ctrl Escenario Basevs Convencional c/Falla					Control CRC-H vs Escenario Base					Control CRC-H vs Convencional c/Falla				
		Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados	Trabajos en cola		Utilización		Trabajos Terminados
		Máx	Final	Máx	Final		Máx	Final	Máx	Final		Máx	Final	Máx	Final	
1	1	-4,35	-33,33	-0,19	-0,64	-2,65	7,69	33,33	1,69	3,22	-1,31	12,00	60,00	1,88	3,86	1,35
	2	-7,69	0,00	0,00	-2,68		50,00	69,23	1,75	9,98		55,56	69,23	1,75	12,63	
2	1	0,00	0,00	5,74	3,07	-0,86	4,35	0,00	6,48	5,29	6,77	4,35	0,00	0,75	2,22	7,63
	2	14,29	-50,00	1,37	1,75		28,00	-50,00	2,83	3,11		14,29	0,00	1,46	1,37	
3	1	54,76	71,01	0,05	0,05	-18,95	19,15	37,50	0,05	0,05	-18,95	-39,78	-45,68	0,00	0,00	0,00
	2	-9,09	-11,11	-10,07	-26,95		-9,09	-11,11	-6,87	-13,91		0,00	0,00	3,21	13,54	
Total				-0,44	-3,67	-6,79			1,01	1,33	-3,28			1,45	5,00	3,51



**Figura 42: Resumen de gráficas línea 2, servidores 1 y 2, Escenario 9**

(—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla



**Figura 43: Trabajos terminados en línea 2 y 3, Escenario 9**

## 4.12 Comentarios generales

En la Tabla 41 se resumen las métricas más significativas al momento de comparar el desempeño del control propuesto CRC-H respecto del Control Convencional en todos los escenarios planteados. Se expresan los resultados en términos porcentuales. A continuación se comentan las observaciones más relevantes:

**Utilización total máxima y final.** Se observa que, en general, el Control CRC-H aumenta la utilización de los servidores y por lo tanto disminuye los tiempos de ocio de los mismos respecto al control convencional, además se observa que al finalizar la simulación, la diferencia máxima de utilización se acentúa aún más, pasando de una diferencia de utilización de +4,27% a 10,76% en favor del control CRC-H.

**Cola Máxima de Línea en Falla.** Se observa que, en todos los casos, la cola del servidor en falla se ve disminuida en el control CRC-H respecto del control convencional, llegando a un máximo de 62% menos de cola en el escenario 5.

**Total de Trabajos Terminados.** Se observa que, en general, el Control CRC-H aumenta la cantidad de trabajos terminados respecto al control convencional, llegando a una máxima diferencia 2,57% trabajos terminados adicionales en el escenario 6.

**Tabla 41: Tabla resumen Métricas de comparación porcentual Control CRC-H vs Convencional**

Control CRC-H vs Convencional	Escenario								
	1	2	3	4	5	6	7	8	9
Utilización Total Máx	+2,51	-0,06	+3,44	+3,23	+2,77	+4,27	+0,97	+2,23	+1,45
Utilización Total Final	+6,66	-0,06	+7,00	+8,59	+8,46	+10,76	+3,07	+5,73	+5
Cola Máx de línea en falla	-47,73	-49,09	-50	-45,45	-62	-51,94	-14,89	-39,62	-39,78
Total de Trabajos Terminados	+0,52	+1,28	0	-2,09	+1,91	+2,57	+2,31	+0,63	+0,51

## CAPÍTULO 5

### Conclusiones

La arquitectura propuesta fue presentada con el fin de encontrar un sistema de control alternativo a las soluciones convencionales aplicadas a sistemas productivos de manufactura capaz de optimizar la respuesta ante perturbaciones en el sistema. Las conclusiones luego de obtener los resultados de los experimentos realizados son las siguientes:

1. Se comprueba que la hipótesis planteada es correcta dado que el sistema de control propuesto tiene, en general, mejor performance respecto de un sistema de control convencional. El aumento de la utilización de los servidores indica una disminución en tiempos ociosos de las máquinas y por lo tanto aumento de la producción en una misma unidad de tiempo, mientras que también permite balancear la carga de trabajos entre las líneas paralelas, esto provoca una disminución de los largos de colas lo cual implica una disminución de la capacidad de almacenamiento requerido en cada línea. No obstante, es necesario investigar y profundizar aún más esta arquitectura de manera que sea capaz de aprovechar todas las propiedades de los sistemas holónicos como la Cooperación, Recursividad y la flexibilidad de crear/eliminar holones.
2. En términos más específicos, se comprueba la eficacia de un sistema de control híbrido basado en un enfoque holónico aplicado a procesos simulados a través de una red de cola, y en particular de redes de Jackson demostrando ser una opción real para controlar un proceso de manufactura. El hecho de que el sistema de control propuesto se comporte como un control convencional cuando está estable (sin perturbaciones) asegura la optimización global del sistema productivo, por otro lado, cambiar a un sistema de control distribuido al detectar una falla (perturbación) asegura mantener una optimización que si bien no es global, es entorno al radio de los puntos de ruteo hacia las líneas paralelas. Es en este último escenario donde se observan las mayores ventajas del sistema de control propuesto respecto al sistema de control convencional ya que este no reacciona ante la falla lo cual produce un desborde de la cola de la línea en falla, no así el sistema de control propuesto que realiza el ruteo de trabajos en función del largo de cola de cada línea, manteniendo las colas relativamente uniformes a pesar de que existe una cola línea en falla.

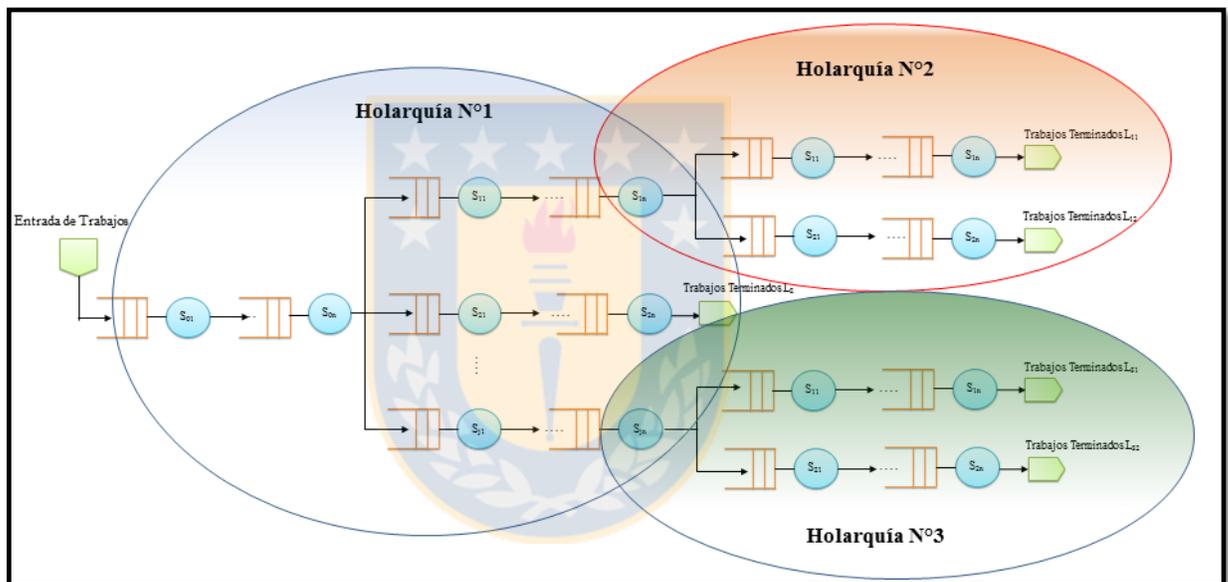
3. Una observación importante es que el sistema de control propuesto aumenta la cantidad de trabajos terminados en una o más de las líneas paralelas que no están en falla respecto al sistema balanceado (sistema de control convencional sin falla) considerado como el mejor escenario. Lo anterior se debe a que al distribuir los trabajos de forma inteligente, el sistema de control propuesto disminuye los tiempos en que las colas de las líneas están vacías, aumentando la utilización de los servidores.
4. Los modelos de redes de cola y, en específico las redes de Jackson, demostraron ser de utilidad para simular procesos, si bien se acotan a una clase de trabajo, la aplicabilidad de este tipo de redes llega desde las redes informáticas con la simulación de transportes de paquetes de datos, simulación de sistemas de conmutación de paquetes en multiprocesadores, sistemas de procesamiento de transacciones, simulación de llegada de un tipo de paciente a un hospital para evaluar congestiones, simulación de llegada de un cliente a un banco, hasta simular procesos de manufactura de múltiples líneas y etapas de producción tales como una producción de tableros, una salmonera, etc.
5. Respecto a la plataforma desarrollada, se comprueba que la integración del toolbox SimEvents de Matlab con JADE a través de MACSimJX resultó ser una excelente herramienta de simulación y de desarrollo de algoritmos de control basados en tecnología multi-agente y, en particular en este caso, en base al paradigma holónico. Esta plataforma integrada permitirá seguir la línea de investigación e incluso abrir nuevas oportunidades a otros paradigmas.

#### Perspectivas y Trabajos Futuros:

Como se comentó en la revisión del estado del arte, este trabajo sirve como base para otras líneas de investigación tales como la aplicación de un sistema de control holónico que embarque la inteligencia del modelo de redes y que reemplazase a un sistema de control convencional en un proceso real o podría ser aplicado en paralelo este para predecir comportamientos futuro dada una falla en el sistema de manera de tomar acciones anticipativas.

Desde el punto de vista del algoritmo de control, la arquitectura propuesta es básica y es el inicio de una línea de investigación que a partir de los resultados obtenidos es prometedora. Sin embargo, es imperativo mejorar el enfoque de la misma, por ejemplo:

- Integrando métodos de negociación entre los holones de línea o directamente entre los holones servidores.
- Expandir la red y con ello la capacidad de cooperar entre holones. Tal como el paradigma holónico, una red puede ser vista como una subred dentro de una red mayor y en la práctica, muchos procesos industriales (y redes informáticas) pueden ser vistos de esta forma; cuentan con más de un punto de derivación y ruta de trabajos, por lo tanto, un enfoque holónico podría ser el considerar un punto de ruteo y sus servidores alrededor como una holarquía (subred desde punto de vista de las redes de cola) la cual puede interactuar con otra holarquía (subred) para lograr un objetivo común. Lo anterior se grafica en la Figura 44.



**Figura 44: Expansión del sistema de control holónico propuesto.  
Elaboración propia.**

Desde el punto de vista de la plataforma desarrollada, demostró ser una poderosa herramienta para probar diferentes paradigmas de control. En particular, la presente tesis abarcó los Sistemas de Control Holónicos, sin embargo, la aplicabilidad de la plataforma puede ser expandida para implementar algoritmos de control multi-agentes, bio-inspirados, sistemas viables, u otro enfoque inteligente. Además, es posible desarrollar una interacción de JADE con otros toolbox de Matlab tales como Control Predictivo, Control Difuso, Redes Neuronales y Control Clásico (Control Proporcional Integral Derivativo) lo que habilita al usuario a embarcar diferentes motores de inteligencia a los agentes desarrollados en JADE.

Desde el punto de vista de proceso, la plataforma permite simular diferentes tipos de procesos discretos a través de la teoría de redes de cola. Considerando lo anterior, una sugerencia de trabajo futuro es expandir las capacidades de la arquitectura de control en sistemas productivos capaces de ser simulados a través de redes BCMP. Lo anterior implicaría que el sistema de control debe ser capaz de soportar múltiples tipos de trabajos. De esta forma, se aumenta una de las características fundamentales requeridas para los nuevos sistemas de manufactura: la flexibilidad.

Como mejoras a la plataforma, se propone:

1. Integrar en la plataforma de Java las gráficas de simulación y la generación automática de reportes. Actualmente los datos son guardados en una matriz en Matlab para luego procesarlos y obtener las gráficas. La implementación de esta propuesta, permitirá disminuir los tiempos de análisis de resultados.
2. Utilizar la reusabilidad del código dada por la tecnología de agentes. Actualmente se utiliza un código diferente para cada holón, sin embargo esto puede ser optimizado y reutilizar código para holones del mismo tipo. La implementación de esta propuesta permitiría disminuir los tiempos de programación de los algoritmos y la depuración de los mismos.

## REFERENCIAS

- [1] P. Leitão, «Agent-based distributed manufacturing control: A state-of-the-art survey,» *Engineering Applications of Artificial Intelligence*, vol. 22, p. 979–991, 2009.
- [2] J. Araújo, R. del-Olmo-Martínez, J. Laviós y J. de-Benito-Martína, «Programación y Control de Sistemas de Fabricación Flexibles - un Enfoque Holónico,» *Revista Iberoamericana de Automática e Informática industrial*, vol. 12, p. 58–68, 2015.
- [3] A. Thomas, D. Trentesaux y P. Valckenaers, «Intelligent distributed production,» *Journal of Intelligent Manufacturing*, vol. 23, p. 2507–2512, 2012.
- [4] L. Monostori, J. Váncza y S. Kumara, «Agent-Based Systems for Manufacturing,» *Procedia CIRP*, vol. 7, p. 437–442, 2013.
- [5] C. Herrera, S. Belmokhtar y A. Thomas, «Viable System Model approach for holonic product-driven manufacturing systems,» de *1st Workshop on Service Orientation in Holonic and Multi Agent Manufacturing Control*, París, 2011.
- [6] P. Leitão, V. Mařík y P. Vrba, «Past, Present, and Future of Industrial Agent Applications,» *IEEE Transactions on industrial Informatics*, vol. 9, n° 4, pp. 2360-2372, 2013.
- [7] J. H. Christensen, «Holonic Manufacturing Systems: Initial architecture and standards directions,» de *First European Conference on Holonic Manufacturing Systems, Hannover, Alemania*, 1994.
- [8] T. Borangiu, S. Raileanu, D. Trentesaux, T. Berger y I. Iacob, «Distributed manufacturing control with extended CNP interaction of intelligent products,» *Journal of Intelligent Manufacturing*, vol. 25, n° 5, pp. 1065-1075, 2013.
- [9] T. Borangiu, S. Răileanu, T. Berger y D. Trentesaux, «Switching mode control strategy in manufacturing execution systems,» *International Journal of Production Research*, vol. 53, n° 7, pp. 1950-1963, 2015.

- [10] J.-C. Huet, J.-L. P. Paris, K. Kouiss y M. Gourgand, «A new reengineering methodology for the product-driven system applied to the medication-use process,» *Decision Support Systems*, vol. 55, pp. 599-615, 2013.
- [11] P. Leitão y F. Restivo, «ADACOR: A holonic architecture for agile and adaptive manufacturing control,» *Computers in Industry*, vol. 57, p. 121–130, 2006.
- [12] P. Vrba, P. Tichy, V. Marík, K. Hall, R. Staron, F. Maturana y P. Kadera, «Rockwell Automation's Holonic and Multiagent Control Systems Compendium,» *IEEE Transactions on systems, man, and cybernetics—Part C: Applications and Reviews*, vol. 41, n° 1, pp. 14-30, 2011.
- [13] WBG, «Manufacturing, value added (current US\$),» 2015. [En línea]. Available: [http://data.worldbank.org/indicator/NV.IND.MANF.CD/countries/1W?order=wbapi\\_data\\_value\\_2010%20wbapi\\_data\\_value&sort=desc&display=default](http://data.worldbank.org/indicator/NV.IND.MANF.CD/countries/1W?order=wbapi_data_value_2010%20wbapi_data_value&sort=desc&display=default). [Último acceso: 03 Agosto 2015].
- [14] NAM, «Facts about manufacturing,» 13 Noviembre 2014. [En línea]. Available: <http://www.nam.org/Newsroom/Facts-About-Manufacturing/>. [Último acceso: 3 Agosto 2015].
- [15] EC, «ManuFuture, A vision for 2020. Report of the High-Level Group,» 2004.
- [16] EC, «ManuFuture, Strategic Research Agenda. Report of the High-Level Group,» 2006.
- [17] D. Trentesaux, «Distributed control of production systems,» *Engineering Applications of Artificial Intelligence*, vol. 22, p. 971–978, 2009.
- [18] B. Render y J. Heizer, Dirección de la Producción y Operaciones - Decisiones Tácticas, 8va ed., Madrid: Pearson, 2007.
- [19] J. Martinez y A. Colombo, «Engineering framework for agent-based manufacturing control,» *Engineering Applications of Artificial Intelligence*, vol. 19, p. 625–640, 2006.
- [20] S. Bussmann y D. McFarlane, «Rationales for Holonic Manufacturing Control,» de *In Proc. of the Second International Workshop on Intelligent Manufacturing Systems*,

1999.

- [21] P. Leitão, «An Agile and Adaptive Holonic Architecture for Manufacturing Control,» 2004.
- [22] M. Duncan y S. Bussmann, «Developments in Holonic Production Planning and Control,» *Production Planning & Control: The Management of Operations*, vol. 11, n° 6, pp. 552-536, 2000.
- [23] P. Leitão, «An agile and adaptive holonic architecture for manufacturing control,» 2004.
- [24] J. Barbosa, P. Leitão, E. Adam y D. Trentesaux, «Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution,» *Computers in Industry*, vol. 66, p. 99–111, 2015.
- [25] P. Mella, *The Holonic Revolution: Holons, Holarchies and Holonic Networks: The Ghost in the Production Machine*, Pavia: Scientifica, 2009.
- [26] A. Koestler, *The Ghost in the Machine*, London: Arkana Books, 1969.
- [27] A. Giret y V. Botti, «Holons and agents,» *Journal of Intelligent Manufacturing*, vol. 15, pp. 645-659, 2004.
- [28] D. McFarlane y S. Bussman, «Holonic Manufacturing Control: Rationales, Developments and Open Issues,» *Agent-Based Manufacturing - Advances in the Holonic Approach*, pp. 303-326, 2003.
- [29] M. Jovanovic, S. Zupatan, M. Starbek y I. Prebil, «Virtual approach to holonic control of the tyre-manufacturing system,» *Journal of Manufacturing Systems*, vol. 33, p. 116–128, 2014.
- [30] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts y P. Peeters, «Reference architecture for holonic manufacturing systems: PROSA,» *Computers in Industry*, vol. 37, p. 255–274, 1998.
- [31] J.-L. Chim y D. McFarlane, «A Holonic Component-Based Approach to Reconfigurable

- Manufacturing Control Architecture,» de *Database and Expert Systems Applications. Proceedings. 11th International Workshop on*, Londres, 2000.
- [32] P. Vrba y V. Mařík, «Simulation in agent-based control systems: MAST case study,» *International Journal of Manufacturing Technology and Management*, vol. 8, 2006.
- [33] W. Shen, Q. Hao, H. Joong Yoon y D. Norrie, «Applications of agent-based systems in intelligent manufacturing: An updated review,» *Advanced Engineering Informatics*, vol. 20, pp. 415-431, 2006.
- [34] J. Hodík, P. Becvar, M. Pechoucek, J. Vokřínek y J. Pospíšil, «ExPlanTech and ExtraPlant: multi-agent technology for production planning, simulation and extra-enterprise collaboration,» *International Journal of Computer Systems Science & Engineering*, vol. 5, pp. 357-367, 2005.
- [35] F. Maturana, D. Norrie y W. Shen, «MetaMorph: An adaptive agent-based architecture for intelligent manufacturing,» *International Journal of Production Research*, vol. 37, nº 10, pp. 2159-2173, 1999.
- [36] W. Shen, F. Maturana y D. H. Norrie, «MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing,» *Journal of Intelligent Manufacturing*, vol. 11, pp. 237-251, 2000.
- [37] P. Valckenaers, D. Mcfarlane y T. Heikkilä, «Towards a Novel Manufacturing Control Principle,» de *The Second International Workshop on Intelligent Manufacturing Systems 1999*, Leuven, 1999.
- [38] A. Singh, D. Juneja y A. Sharma, «Agent Development Toolkits,» *International Journal of Advancements in Technology*, vol. 2, nº 1, pp. 158-164, 2011.
- [39] G. Bolch, S. Greiner, H. de Meer y K. Trivedi, *Queueing Networks and Markov Chains*, New York: John Wiley & Sons, Inc., 1998.
- [40] C. R. Robinson, P. Mendham y T. Clarke, «MACSimJX: A Tool for Enabling Agent Modelling with Simulink Using JADE,» *Journal of Physical Agents*, vol. 4, nº 3, pp. 1-

7, 2010.

- [41] A. Lüder, J. Peschke y T. Sauter, «Distributed intelligence for plant automation based on multi-agent systems: the PABADIS approach,» *Production Planning & Control: The Management of Operations*, vol. 15, n° 2, pp. 201-212, 2004.



# ANEXO 1

## Plataforma de Desarrollo

El presente anexo consiste en detallar como realizar la puesta en marcha de la plataforma desarrollada integrada por Matlab, JADE y MACSimJX.

### 3.3.4.1 A1.1 Conocimientos previos

Se aconseja realizar el estudio previo de las siguientes herramientas:

- ✓ Matlab (incluido Simulink)
- ✓ Lenguaje de Programación JAVA y de un entorno de desarrollo como NetBeans.
- ✓ JADE y su integración a un entorno de desarrollo JAVA.

### 3.3.4.2 A1.2 Versión de Software a utilizar

MACSimJX es el software (middleware) que realiza el traspaso de información bidireccional entre la plataforma en donde se simula la red de cola (Matlab) y el software en donde viven los agentes u holones (JADE) que ejecutan los algoritmos de control y por lo tanto es el software que determina con qué versión de Windows, Matlab y JADE es compatible. El primer paso es descargar MACSimJX desde la página oficial del software: <http://www.agentcontrol.co.uk/> y según la versión que se descargue (actualmente existe para Matlab 2009 y Matlab 2010), instalar Matlab, por otro lado, es importante considerar que a la fecha la única versión de JADE soportada por MACSimJX es JADE 3.6.1, la cual puede ser descargada de las misma página indicada anteriormente o desde la página oficial de JADE <http://jade.tilab.com/> .

### **3.3.4.3 A1.3 Recomendaciones de Instalación de Software**

Se recomienda comenzar realizando la instalación de Matlab y JADE para posteriormente realizar la instalación de MACSimJX. La instalación MACSimJX se realiza siguiendo paso a paso las instrucciones del documento README.pdf que se descarga en conjunto con el software.

### **3.3.4.4 A1.4 Funcionalidades principales de MACSimJX**

Como se ha comentado anteriormente, MACSimJX es un software que crea un túnel de comunicación entre el Modelo Simulink de Matlab y JADE. Para realizar esto, MACSimJX cuenta con una S-function para crear una canal de comunicación de información entre Simulink y un programa externo con capacidad de procesamiento en paralelo que caracteriza a un sistema multiagente, en este caso JADE bajo una arquitectura cliente-servidor.

MacsimJX tiene dos componentes fundamentales; el Agent Environment y la Agent Task Force. El Agent Environment (AE) es el ambiente que permite:

- Mantener el registro de los agentes actuales y facilitar el nacimiento y muerte dinámica de los agentes. Lo anterior es manejado por el Agente Directory Facilitator (DF) de JADE.
- Sincronización con simulink a través de macsim.
- Proveer la entrada actual y el paso de tiempo cuando es requerido.
- Guardar datos a ser enviados de vuelta a simulink.
- Tener la capacidad de difundir los mensajes a la comunidad de agentes.

Para lo anterior, el AE se vale del Agent Server (AS) y Agent Coordinator (AC). El AS tiene la finalidad de establecer el canal con simulink y actúa como un intérprete entre simulink y el agentCoordinator. Además, el AS es responsable de solicitar a todos los agentes cerrarse cuando el usuario decide terminar el programa. El AC por su parte provee de actualizaciones a los agentes cuando llega un nuevo dato. También pone en marcha al Agent Server y entonces se inicializa el grupo de agentes, mantiene los enlaces de comunicación con agentes que deseen recibir datos entrantes. Cuando el AC recibe un arreglo de datos desde simulink, este lo difunde a los agentes registrados para este servicio. Cuando estos agentes han terminado de

trabajar con los datos, ellos retornan los datos al AC el cual pasa los datos procesados a simulink (Ver Figura 45).

El segundo componente relevante de MACSimJX es el Agent Task Force (ATF), consiste en todos los agentes que operan en forma conjunta sobre de los datos que arriban desde simulink con el objetivo de cumplir con alguna tarea. En términos prácticos, es un directorio en donde se encuentran todos los códigos fuentes y códigos compilados de la aplicación.

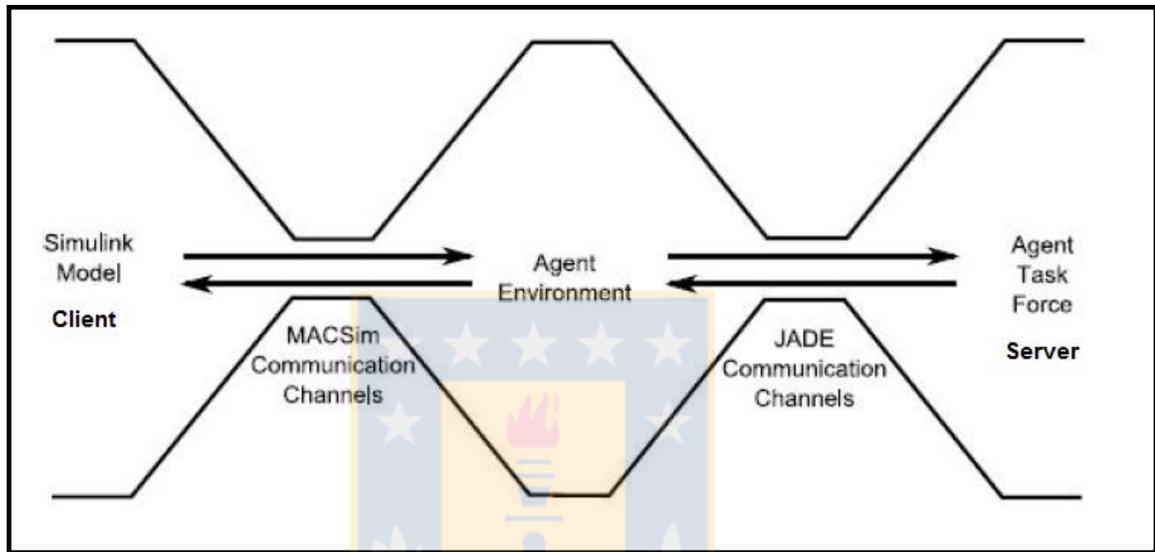


Figura 45: Comunicación Simulink/JADE, imagen adaptada de [40]

### 3.3.4.5 A1.5 Puesta en Marcha de MACSimJX

Para comenzar, se recomienda correr el programa demo prediseñado. Dentro de la carpeta ATF existe la carpeta “example” la cual ya contiene los códigos fuentes de una aplicación, además, dentro de la carpeta “SimModel”, se encuentra el archivo del sistema a simular “example.mdl”. Una vez instalado todo el software se recomienda poner en marcha este ejemplo a través de las instrucciones indicadas en el documento README.pdf para verificar que se establece la comunicación Matlab-JADE. Luego, para los proyectos futuros, se recomienda utilizar como base de trabajo tanto los archivos fuente de la carpeta example como el archivo \*.mdl.

### 3.3.4.6 A1.5 Personalización del ATF

A continuación se realiza una descripción secuencial de la estructura del código fuente de los agentes del ATF para usarlo como base (plantilla) para generar los trabajos personalizados:

1. Declaración de package relacionados.
2. Declaración de librerías.
3. Inicio de clase Agente (*public class Nombre\_Agente extends UsefulAgentMethods*).
  - 3.1. Definición de los puertos de entrada y salida a monitorear/controlar por este agente.
  - 3.2. Declaración de variables.
  - 3.3. Inicialización del agente (*Setup()*).
    - 3.3.1. Llamado del comportamiento del agente.
    - 3.3.2. Set y registro de servicios a suministrar por el agente.
    - 3.3.3. Adquisición del número asignado para este agente a través de la interfaz de MACSimJX.
    - 3.3.4. Identificación del Agente Coordinador (AC).
    - 3.3.5. Identificación de los agentes de un servicio de interés de este agente.
  - 3.4. Fin de Inicialización
  - 3.5. Declaración del comportamiento cíclico.
    - 3.5.1. Declaración de variables.
    - 3.5.2. Recepción y procesamiento de mensajes desde Simulink.
    - 3.5.3. Se envían datos a otros agentes.
    - 3.5.4. Recepción y procesamiento de mensajes desde otros agentes.
    - 3.5.5. Envío de datos a Simulink.
  - 3.6. Fin de Comportamiento.
  - 3.7. Declaración de Función para enviar datos a Simulink (*finishCalculationsThenSend()*).
    - 3.7.1. Almacenamiento de información en el arreglo para enviar datos a simulink (*dataOut*).
  - 3.8. Fin de Función para enviar datos a Simulink.
  - 3.9. Termina el agente (*takeDown*).
4. Fin de Clase Agente.

Los códigos fuentes de este proyecto de tesis en particular se encuentra en el Anexo 2.

### 3.3.4.7 A1.5 Puesta en Marcha de la Plataforma Desarrollada

A continuación se indicarán los pasos para la puesta en marcha de la plataforma desarrollada.

Una vez instalado MACSimJX en C:\ debe realizar lo siguiente:

1. **Preparación de archivos.** Para realizar esta etapa es necesario contar con los archivos adjuntos en el CD de la tesis.
  - a. Copiar la carpeta “procesoDescentralizado” en la ruta C:\macsimjx\ATFs\src. Esta carpeta contiene todos los códigos fuente del sistema de control desarrollado.
  - b. Copiar el archivo ejecutable Inicio.bat en la ruta C:\macsimjx. Este ejecutable fue programado para ejecutar JADE y posteriormente MACSimJX de forma automática con doble click.
  - c. Copiar el archivo modelo procesoDescentralizado.mdl en la ruta C:\macsimjx\SimModel.
  - d. Ejecutar el archivo procesoDescentralizado.mdl.
2. **Compilación de código fuente.** Dentro de la carpeta “procesoDescentralizado” existe un archivo ejecutable Compilación.bat el cual fue programado para compilar de forma automática los códigos fuentes que se encuentren en la ruta de este ejecutable.
3. **Inicialización de la aplicación.** Se realiza doble click en el archivo ejecutable Inicio.bat, lo cual desplegará la interfaz de MACSimJX (Figura 46) y JADE (Figura 47). En ese momento JADE se encontrará ejecutando el Agent Coordinator de MACSimJX.
4. **Configuración Inicial.** Dado que MACSimJX no reconoce automáticamente al ATF es necesario configurarlo la primera vez que se ejecuta la aplicación. A continuación se describen las celdas de la interfaz de MACSimJX para configurar el ATF luego de presionar *Add Option*:
  - a. *ATF Name* : se ingresa un nombre cualquiera del ATF, ejemplo: “procesoDescentralizado”.
  - b. *MACSim Input* : se ingresa la cantidad de entradas que tendrá el sistema, este número es coincidente con el número de entradas de la S-Function en el modelo Simulink.

- c. *MACSimOutput* : se ingresa la cantidad de salidas que tendrá el sistema, este número es coincidente con el número de salidas de la S-Function en el modelo Simulink.
  - d. *Sample Rate/Hz* : se ingresa la tasa de muestreo del sistema.
5. **Identificación de cada agente (holón) en el ATF.** MACSimJX necesita que cada agente sea identificado por un nombre único, la ruta del código fuente y un número único. Por cada holón agregado debe presionar *Add agent to ATF*. En particular, esta aplicación tenemos 4 holones; 3 holones de línea y 1 de ruteo. Luego de finalizar, se debe crear el ATF pulsando *Add ATF to List*.
6. **Finalización de la configuración.** Creado el ATF, presionar *Continue*, automáticamente se creará el Agent Server y los Holones ATF en JADE (Ver Figura 48), y se ejecutarán las GUI de cada holón creado (Ver Figura 49).

Con los pasos anteriores la plataforma ya está operativa para simular el proceso. Las posibilidades de las GUI están definidas en el capítulo 4.

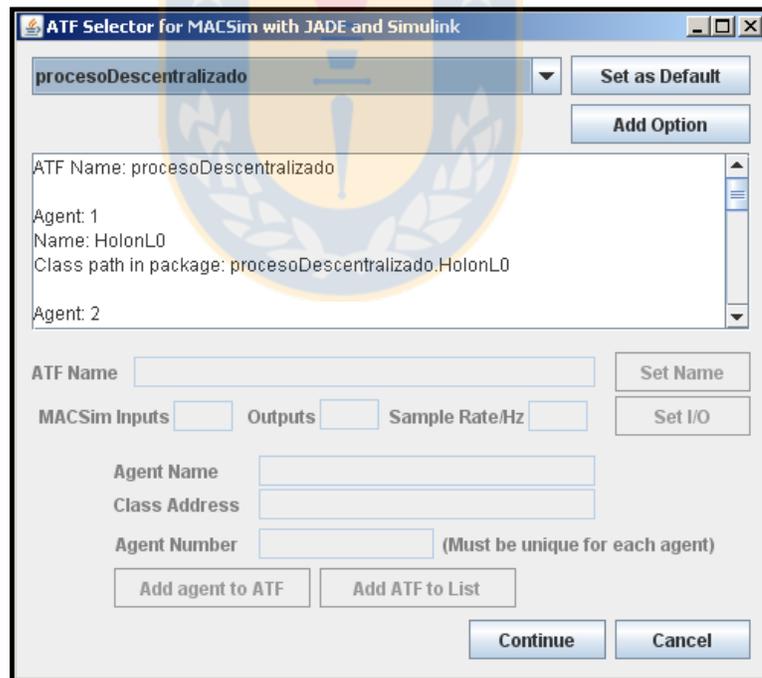


Figura 46: Interfaz gráfica MACSimJX

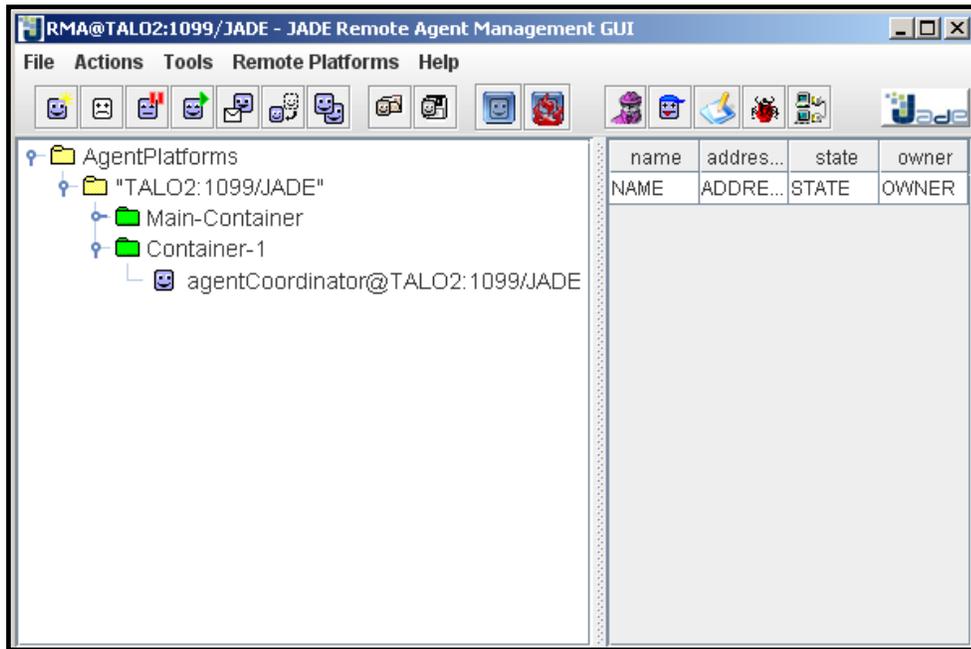


Figura 47: Interfaz de JADE ejecutando el Agent Coordinator de MACSimJX

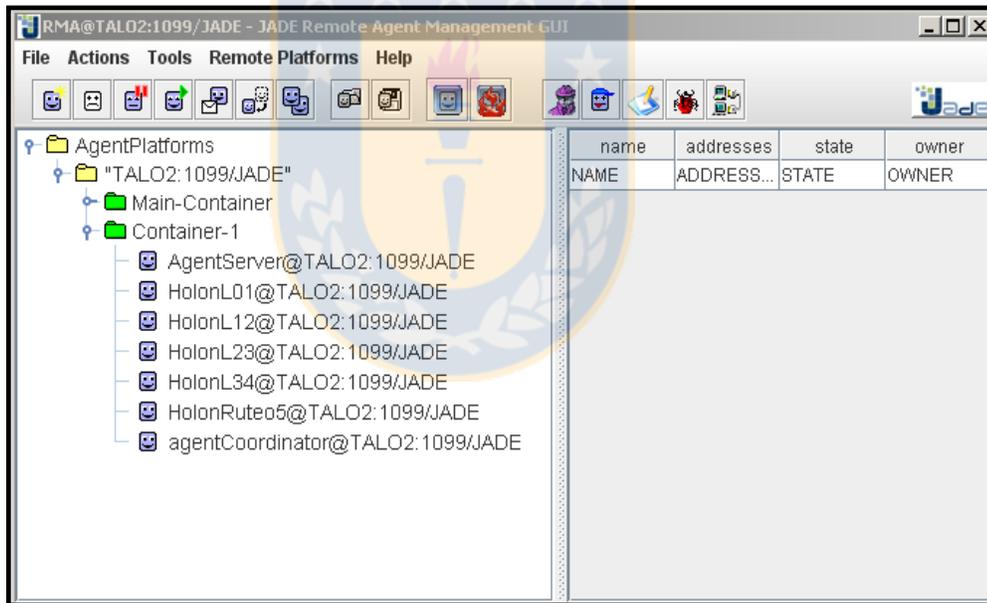


Figura 48: Interfaz Gráfica JADE con MACSimJX y Holones de aplicación en ejecución

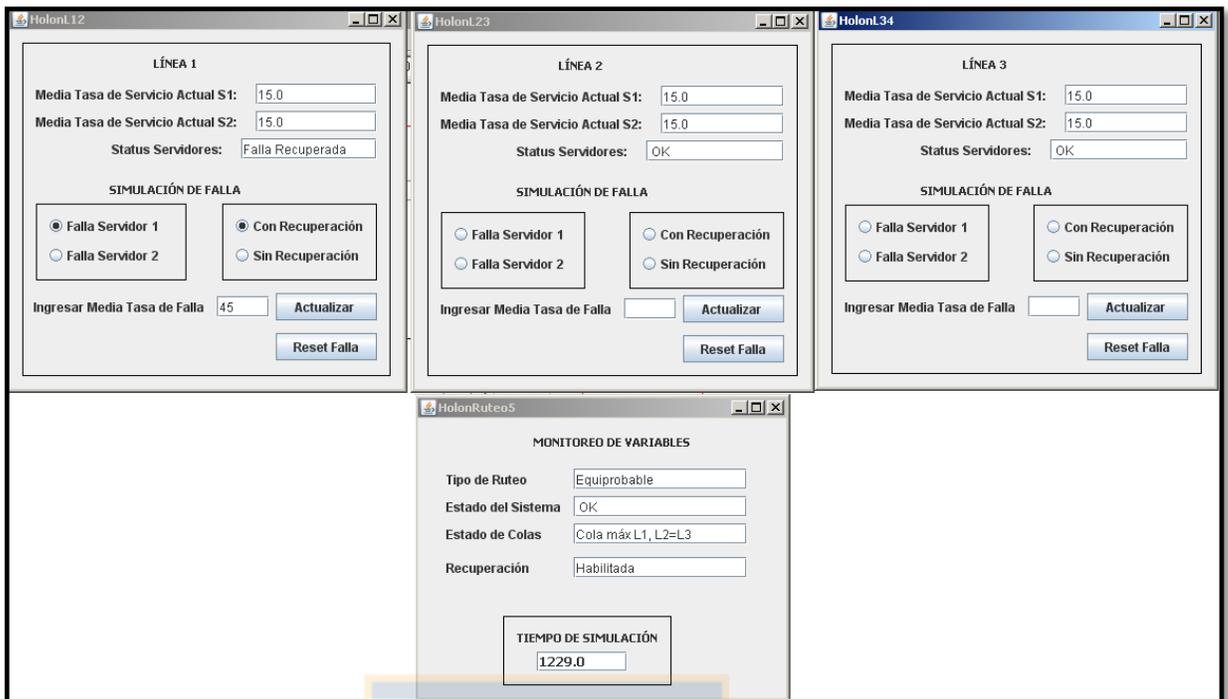


Figura 49: Holones de Línea y Ruteo en operación y comunicados con Simulink



## ANEXO 2

### Códigos Fuente

#### 3.3.4.8 A.2.1 Código Fuente Holón de Ruteo

```
/**dsdsd
 * @author Italo
 */
import macsimjx.*;

import jade.lang.acl.ACLMessage;
import jade.domain.FIPAAException;
import jade.domain.DFService;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.Agent;
import jade.core.AID;
import misagentes.GuiHolonRuteo;
import java.util.Random;

public class HolonRuteo extends UsefulAgentMethods {
    Agent thisAgent = (Agent) this;
    int agentNumber = 0;

    private GuiHolonRuteo myGui;

    /*
     * These determine which elements of the input data that this agent
     * is interested in, and which element of the out going data it is
     * changing.
     */
    /***DEFINICIÓN DE PUERTOS E/S DE INTERÉS DESDE/HACIA SIMULINK***/
    int[] inputPortsOfInterest = {0,8}; // I.e. Top signal input to MACSim
    block.
        //0=Cola serv. 1 (Sin utilidad)
        //8=Tiempo de Simulación
    int[] outputPorts = {0,1,2,3}; // Set which ports in Simulink data is
    shown on.
        //0 = ruta;
        //1 = probL1;
        //2 = probL2;
        //3 = probL3;

    /***DECLARACIÓN DE VARIABLES***/
    // Set to true for helpful output.
    boolean debug = false;

    // Keep track of number of information agents.
    private int agentsSubscribed = 0;
    private int agentsLeftToRespond = 0;

    boolean firstRun = true;
    boolean dataUpdated = false;
    AID[] agentIDs = null;
    AID agentID = null;
```

```

AID AgentCoordinatorID = null;

TimeStepData tsd = new TimeStepData ();

/**DECLARACIÓN DE VARIABLES PERSONALIZADAS***/
double colaS1L0=0;
double colaS2L0=0;
double colaS1L1=0;
double colaS2L1=0;
double colaS1L2=0;
double colaS2L2=0;
double colaS1L3=0;
double colaS2L3=0;

/**VARIABLES PARA GENERAR NÚMEROS ALEATORIOS***/
float random=0;

Random rnd = new Random ();
/*******
int ruta=1;
double probL1=0.334;
double probL2=0.333;
double probL3=0.333;

double[] colas = {0,0,0};
double colamax =3;
double colamin =1;
double colamedia =2;

/*******
double probabilidadColaMax = 0.1;
double probabilidadColaMed = 0.2;
double probabilidadColaMin = 0.7;
/*******
double probabilidadadmin = 1;
double fallaL0 =0;
double fallaL1 =0;
double fallaL2 =0;
double fallaL3 =0;
double recuperacionL1 =0;
double recuperacionL2 =0;
double recuperacionL3 =0;
double resumenFalla =0;
double mediafalla=15.0;
double tiemposimulacion=0.0;

public void setup() {

    // ****CREACIÓN Y DESPLIEGUE DE LA HMI DEL AGENTE****
myGui = new GuiHolonRuteo(this);
myGui.setVisible(true);
//Se setea semilla
rnd.setSeed(200);
    addBehaviour(new filterBehaviour());

    /*
     * Register agent with directory facilitator of JADE along with the

```

```

        * services it provides (in this case, arithmetic and generic agent
services).
        * NB - To change properties after registering use the JADE
'modify' method.
        */
String[] services = {"arithmetic", "Agent"};
registerAgent(thisAgent, services, services);

// Get agent number from argument provided by instantiator of this
agent.
Object[] args = getArguments();
if (args != null && args.length > 0) {
    String firstArg = args[0].toString();
    agentNumber = new Integer(firstArg).intValue();

} else {
    // Terminate agent.
    System.out.println("No agent id");
    doDelete();
}

try {
    // Locate data provider.
    AgentCoordinatorID = getAgentIDsOfService(thisAgent,
        "InputsUpdate")[0];

    // Find agents that are interested in this agent's data.
    agentIDs = getAgentIDsOfService(thisAgent, "arithmetic");

    /*
    * N.B. If environment is dynamic would need to perform these
    * searches either more than once to check for new agents, or
    * implement some service subscription method to receive a
message
    * when new agents enter environment.
    */

} catch (Exception e) {
    System.out.println("Error finding service: " + e);
}

agentsSubscribed = countSubscribingAgents(thisAgent, "arithmetic");

if (debug)
    System.out.println("agentsSubscribed to arithmetic service:"
        + agentsSubscribed);

agentsLeftToRespond = agentsSubscribed - 1;

}

```

```

//*****DECLARACIÓN DEL COMPORTAMIENTO CICLICO DEL HOLÓN*****
class filterBehaviour extends CyclicBehaviour {
    public void action() {

        TimeStepData tsd = new TimeStepData();
        double[] dataArray;
        int length;

        // *****RECEPCIÓN DE MENSAJES*****
        ACLMessage msg = receive();

        if (msg != null) {
            String message = msg.getConversationId();

            //***** SE FILTRA EL MENSAJE POR LA ID DE LA CONVERSACIÓN
            //**** UPDATEDATA SIGNIFICA QUE VIENE DEL AGENTE COORDINADOR (ES DECIR,
            SIMULINK MATLAB)
                // If new data from AgentCoordinator (ie Simulink).
                if (message.equals("UpdateData")) {
                    if (debug) {
                        System.out.println("(Debug)Agent receiving new data
from Simulink. ");
                        System.out.printf("\n");
                    }
                    try {
                        tsd = (TimeStepData) msg.getContentObject();
                    } catch (Exception e) {
                        System.out.println("Exception with content object"
+ e);
                    }
                //***GENERACIÓN DEL SIGUIENTE NÚMERO ALEATORIO PARA VARIABLES ALEATORIAS***
                random=rnd.nextFloat();
                //**** Se extraen lo datos del arreglo que llega desde
                simulink

                // Extract data from incoming array.
                length = tsd.getData().length;
                dataArray = new double[length];
                for (int i = 0; i < length; i++)
                    dataArray[i] = tsd.getData()[i];

                if (debug) {
                    System.out.println("(Debug)Incoming data: ");
                    for (int j = 0; j < length; j++) {
                        System.out.print(j + ": " + dataArray[j] + "
");
                    }
                    System.out.printf("\n");
                }
            }
        }
    }
}

```

```

//***** ALGORITMO DE CONTROL DEL HOLÓN*****

// ***** SE EXTRAEN LOS DATOS RELEVANTE DEL PUERTO DE INTERÉS

// Se extraen las colas de los servidores 1 de la Línea
1,2 y 3

tiemposimulacion = (double)
dataArray[inputPortsOfInterest[1]];

myGui.campoTiempoSimulacion.setText(Double.toString(tiemposimulacion));

colas[0] = colaS1L1;
colas[1] = colaS1L2;
colas[2] = colaS1L3;

// *** MONITOREO DEL ESTADO DEL SISTEMA
if(fallaL1==0 && fallaL2==0 && fallaL3==0)
{myGui.campoEstadoSistema.setText(" OK ");
resumenFalla =0;
}
else if(fallaL1==1)
{myGui.campoEstadoSistema.setText("Falla en Línea 1");
resumenFalla =1;
}
else if(fallaL2==1)
{myGui.campoEstadoSistema.setText("Falla en Línea 2");
resumenFalla =1;
}
else if(fallaL3==1)
{myGui.campoEstadoSistema.setText("Falla en Línea 3");
resumenFalla =1;
}
if(recuperacionL1==0 && recuperacionL2==0 &&
recuperacionL3==0)
{
myGui.campoRecuperacion.setText("Deshabilitada");
}
else if(recuperacionL1==1 || recuperacionL2==1 ||
recuperacionL3==1)
{
myGui.campoRecuperacion.setText("Habilitada");
}

//*****REGLAS DE CONTROL DE TASA*****
//El algoritmo verifica la cola más larga, más pequeña y la
media en los servidores 1 de las
// líneas L1, L2 y L3. Aigna la probabilidad más alta a la cola
más pequeña,
// la probabilidad más baja a la cola más larga y,
// la probabilidad media a la colas media.
if(fallaL1==1 || fallaL2==1 || fallaL3==1)
{ //TODO ESTE CÓDIGO SE EJECUTA SI Y SOLO SI EXISTE UNA FALLA

//Si la recuperación es 1 y tiempo menor a 1200 ó
recuperación es 0
myGui.campoTipoRuteo.setText("Inteligente");
}

```

```

if(colas[0]>colas[1] && colas[1]>colas[2])
{
    probL1=probabilidadColaMax;
    probL2=probabilidadColaMed;// (Indice=1)
    probL3=probabilidadColaMin;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L1, min L3");
}
else if(colas[0]>colas[2] && colas[2]>colas[1])
{
    probL1=probabilidadColaMax;// (Indice=0)
    probL3=probabilidadColaMed;// (Indice=1)
    probL2=probabilidadColaMin;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L1, min L2");
}
else if(colas[0]>colas[2] && colas[2]==colas[1])
{
    probL1=probabilidadColaMax;// (Indice=0)
    probL3=0.45;// (Indice=1)
    probL2=0.45;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L1, L2=L3");
}
else if(colas[0]==colas[2] && colas[2]>colas[1])
{
    probL1=0.15;// (Indice=0)
    probL3=0.15;// (Indice=1)
    probL2=probabilidadColaMin;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L1=L3, min L2");
}
else if(colas[0]==colas[1] && colas[1]>colas[2])
{
    probL1=0.15;// (Indice=0)
    probL3=0.15;// (Indice=1)
    probL2=probabilidadColaMin;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L1=L2, min L3");
}
else if(colas[1]>colas[0] && colas[0]>colas[2])
{
    probL2=probabilidadColaMax;// (Indice=0)
    probL1=probabilidadColaMed;// (Indice=1)
    probL3=probabilidadColaMin;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L2, min L3");
}
else if(colas[1]>colas[2] && colas[2]>colas[0])
{
    probL2=probabilidadColaMax;// (Indice=0)
    probL3=probabilidadColaMed;// (Indice=1)
    probL1=probabilidadColaMin;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L2, min L1");
}
else if(colas[1]>colas[2] && colas[2]==colas[0])
{
    probL2=probabilidadColaMax;// (Indice=0)
    probL3=0.45;// (Indice=1)
    probL1=0.45;// (Indice=2)
myGui.campoEstadoColas.setText("Cola máx L2, min =L3");
}

```

```

    }
    else if (colas[1]==colas[2] && colas[2]>colas[0])
    {
        probL2=0.15;// (Indice=0)
        probL3=0.15;// (Indice=1)
        probL1=probabilidadColaMin;// (Indice=2)
        myGui.campoEstadoColas.setText("Cola máx L2=L3, min L1");
    }
    else if (colas[2]>colas[1] && colas[1]>colas[0])
    {
        probL3=probabilidadColaMax;// (Indice=0)
        probL2=probabilidadColaMed;// (Indice=1)
        probL1=probabilidadColaMin;// (Indice=2)
        myGui.campoEstadoColas.setText("Cola máx L3, min L1");
    }
    else if (colas[2]>colas[0] && colas[0]>colas[1])
    {
        probL3=probabilidadColaMax;// (Indice=0)
        probL1=probabilidadColaMed;// (Indice=1)
        probL2=probabilidadColaMin;// (Indice=2)
        myGui.campoEstadoColas.setText("Cola máx L3, min L2");
    }
    else if (colas[2]>colas[0] && colas[0]==colas[1])
    {
        probL3=probabilidadColaMax;// (Indice=0)
        probL1=0.45;// (Indice=1)
        probL2=0.45;// (Indice=2)
        myGui.campoEstadoColas.setText("Cola máx L3, L2=L1");
    }
    else if (colas[2]==colas[0] && colas[0]==colas[1])
    {
        probL3=0.334;// (Indice=0)
        probL1=0.333;// (Indice=1)
        probL2=0.333;// (Indice=2)
        myGui.campoEstadoColas.setText("Iguales");
    }
    else
    {myGui.campoEstadoColas.setText("No se cumple condición");
    }

    //LLave del if falla=false o true
}
else if (recuperacionL1==1 || recuperacionL2==1 ||
recuperacionL3==1 && tiemposimulacion>=1200)
{
    myGui.campoTipoRuteo.setText("Equiprobable");

    probL1=0.334;
    probL2=0.333;
    probL3=0.333;

}

```

```

// ***** RUTEO DE TRABAJOS *****

        if (random<probL1) //Probabilidad de que el trabajo se
dirija hacia Línea 1
        {ruta=1;}
        else if (random<=(probL1+probL2)) //Probabilidad de que el
trabajo se dirija hacia Línea 2
        {ruta=2;}
        else //Probabilidad de que el trabajo se dirija hacia Línea 3
        {ruta=3;}

        /* Perform here any calculations required before using
data from
        * other agents.
        */

/**          Block          ends          here
*****/
        if (debug) {
            System.out.println(getLocalName() +
                " awaiting info from "
                + agentsLeftToRespond + "
agent(s)");
        }
        dataUpdated = true;

        /*
        * Pass information about the new data to the
        * other agents.
        */
        if (agentsSubscribed > 1) {
            try {

/** Set information in dataStructure to be passed to other agents
*****/
// ***INFORMACIÓN ENVIADA AL RESTO DE LOS HOLONES DE LA HOLARQUÍA***
//Sin utilidad para e algortimo diseñado
                double [] dataToShare = {resumenFalla};
                int [] elementToChange ={0};
                TimeStepData tsdInner= new TimeStepData ();

                // Populate data structure with required data to exchange.
                tsdInner.setData(dataToShare);
                tsdInner.setElementsToChange(elementToChange);

            }

/**          Block          ends          here
*****/

            // Send information off to other agents.
            for (int i = 0; i < agentIDs.length; i++) {
                if (!agentIDs[i].getLocalName().equals(
                    thisAgent.getLocalName())) {
                    sendObject(agentIDs[i], "agentData",
                        tsdInner);
                }
            }
        } catch (Exception e) {
            System.out.println("Exception: " + e);
        }

```

```

    }
}

    if (agentsLeftToRespond == 0 && dataUpdated) {
        finishCalculationsThenSend();
    }

    //****RECEPCIÓN DE MENSAJES ENVIADOS POR OTROS
HOLONES*****
    // Additional information from another agent.

    //MENSAJES DESDE HOLON L0
    if (message.equals("agentDataL0")
        && !msg.getSender().getLocalName().equals(
            thisAgent.getLocalName())) {
        try {
            tsd = (TimeStepData) msg.getContentObject();
        } catch (Exception e) {
            System.out.println("Exception getting content
object: "
                                + e);
        }

        length = tsd.getData().length;
        dataArray = new double[length];
        for (int i = 0; i < length; i++)
            dataArray[i] = tsd.getData()[i];

    //*****TAREAS ESPECÍFICAS DE ESTE AGENTE*****

    /** Agent specific task inside this block
    *****/
        int [] elementsToUse = tsd.getElementsToChange();
        fallaL0 = dataArray[elementsToUse[0]];
        colaS1L0 = dataArray[elementsToUse[1]];
        colaS2L0 = dataArray[elementsToUse[2]];

        // Perform calculations with data from other agents.

    /** Block ends here
    *****/

        agentsLeftToRespond--;
        if (debug)
            System.out.println(getLocalName() + " recieved info
"
                                + " awaiting "
                                + agentsLeftToRespond + " more.");

        if (agentsLeftToRespond == 0 && dataUpdated) {
            finishCalculationsThenSend();
        }
    }
}

```

```

//*****MENSAJES DESDE HOLON L1*****

    if (message.equals("agentDataL1")
    && !msg.getSender().getLocalName().equals(
        thisAgent.getLocalName())) {
    try {
        tsd = (TimeStepData) msg.getContentObject();
    } catch (Exception e) {
        System.out.println("Exception getting content
object: "
                                + e);
    }

    length = tsd.getData().length;
    dataArray = new double[length];
    for (int i = 0; i < length; i++)
        dataArray[i] = tsd.getData()[i];

    //*****ALMACENAMIENTO DE INFORMACIÓN DESDE L1*****
    /** Agent specific task inside this block
    *****/
        int [] elementsToUse = tsd.getElementsToChange();
        fallaL1 = dataArray[elementsToUse[0]];
        colaS1L1 = dataArray[elementsToUse[1]];
        colaS2L1 = dataArray[elementsToUse[2]];
        recuperacionL1 =dataArray[elementsToUse[3]];

    /** Block ends here
    *****/

    agentsLeftToRespond--;
    if (debug)
        System.out.println(getLocalName() + " recieved info
"
                                + " awaiting "
                                + agentsLeftToRespond + " more.");

    if (agentsLeftToRespond == 0 && dataUpdated) {
        finishCalculationsThenSend();
    }
}

// MENSAJES DESDE HOLON L2
    if (message.equals("agentDataL2")
    && !msg.getSender().getLocalName().equals(
        thisAgent.getLocalName())) {
    try {
        tsd = (TimeStepData) msg.getContentObject();
    } catch (Exception e) {
        System.out.println("Exception getting content
object: "
                                + e);
    }

    length = tsd.getData().length;
    dataArray = new double[length];
    for (int i = 0; i < length; i++)
        dataArray[i] = tsd.getData()[i];

```

```

//*****ALMACENAMIENTO DE INFORMACIÓN DESDE L2*****
/**      Agent      specific      task      inside      this      block
*****/
int [] elementsToUse = tsd.getElementsToChange();

fallaL2 = dataArray[elementsToUse[0]];
colaS1L2 = dataArray[elementsToUse[1]];
colaS2L2 = dataArray[elementsToUse[2]];
recuperacionL2 =dataArray[elementsToUse[3]];

/**      Block      ends      here
*****/

agentsLeftToRespond--;
if (debug)
    System.out.println(getLocalName() + " recieved info
"
                    + " awaiting "
                    + agentsLeftToRespond + " more.");

if (agentsLeftToRespond == 0 && dataUpdated) {
    finishCalculationsThenSend();
}
}

//*****MENSAJES DESDE HOLON L3*****
if (message.equals("agentDataL3")
&& !msg.getSender().getLocalName().equals(
    thisAgent.getLocalName())) {
try {
    tsd = (TimeStepData) msg.getContentObject();
} catch (Exception e) {
    System.out.println("Exception getting content
object: "
                    + e);
}

length = tsd.getData().length;
dataArray = new double[length];
for (int i = 0; i < length; i++)
    dataArray[i] = tsd.getData()[i];

```

```

//*****ALMACENAMIENTO DE INFORMACIÓN DESDE L3*****

/**      Agent      specific      task      inside      this      block
*****/
int [] elementsToUse = tsd.getElementsToChange();

fallaL3 = dataArray[elementsToUse[0]];
colaS1L3 = dataArray[elementsToUse[1]];
colaS2L3 = dataArray[elementsToUse[2]];
recuperacionL3 =dataArray[elementsToUse[3]];

// Perform calculations with data from other agents.

//In this case find the sum of both signals.

/**      Block      ends      here
*****/

agentsLeftToRespond--;
if (debug)
    System.out.println(getLocalName() + " recieved info
"
                    + " awaiting "
                    + agentsLeftToRespond + " more.");

if (agentsLeftToRespond == 0 && dataUpdated) {
    finishCalculationsThenSend();
}
}
//CONFIRMACIÓN DE LA RECEPCIÓN DEL MENSAJE
//*****
if (message.equals("DataAmended")) {
    if (debug)
        System.out.println("Agent " + agentNumber + "
finished.");

    replyToAgent (msg.getSender(), "ProcessingComplete");
}

if (message.equals("Shutting Down")) {
    takeDown (msg.getSender());
}
} else {
    block();
}

}
}

```

```

/**
 * Finalise calculations and return to AgentCoordinator and Simulink.
 */
public void finishCalculationsThenSend() {

/**      Agent      specific      task      inside      this      block
*****
// Carry out any final calculation before returning back to Simulink
and
// specified port.

//***** ENVÍO DE DATOS A SIMULINK*****

double[] dataOut = new double[outputPorts.length];
dataOut[0] = ruta;
dataOut[1] = probl1; //Probl1
dataOut[2] = probl2; //Probl2
dataOut[3] = probl3; //Probl3

/**      Block      ends      here
*****

    tsd.setData(dataOut);
    tsd.setElementsToChange(outputPorts);

    // Send modified data back to Simulink.
    ACLMessage reply = new ACLMessage(ACLMessage.INFORM);
    reply.setConversationId("ProcessedData");

    try {
        reply.setContentObject(tsd);
    } catch (Exception e) {
        System.out.println("Exception setting tsd: " + e);
    }

    if (debug)
        System.out.println(getLocalName() + " finished, sending data to
"
                                + AgentCoordinatorID.getLocalName());
    reply.addReceiver(AgentCoordinatorID);
    send(reply);
    agentsLeftToRespond = agentsSubscribed - 1;
    dataUpdated = false;
}

```

```
/**
 * Terminate agent.
 *
 * @param senderAID AID
 */
protected void takeDown(AID senderAID) {
    try {
        System.out.println("Deregistering and closing " + thisAgent);
        DFService.deregister(thisAgent);
    } catch (FIPAException fe) {
        System.out.println("Problem deregistering " + fe);
    }
    if (senderAID != null)
        contactAgent(senderAID, "Agent Deregistered");

    doDelete();
}
}
```



### 3.3.4.9 A.2.2 Código Fuente de Holón de Línea

```
/**
 *
 * @author Italo
 */

import misagentes.*;
import macsimjx.*;

import jade.lang.acl.ACLMessage;
import jade.domain.FIPAException;
import jade.domain.DFService;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.Agent;
import jade.core.AID;
import java.util.Random;
import misagentes.GuiHolonL1;
import java.util.Random;

public class HolonL1 extends UsefulAgentMethods {
    Agent thisAgent = (Agent) this;
    int agentNumber = 0;

    private GuiHolonL1 myGui;

    /*
     * These determine which elements of the input data that this agent
     * is interested in, and which element of the out going data it is
     * changing.
     */
    int[] inputPortsOfInterest = {2,3,8}; // (Cola serv. 1, Cola serv.2,
    Tiempo de Simulación) I.e. Top signal input to MACSim block.
    int[] outputPorts = {6,7}; // Set which ports in Simulink data is shown
    on.
    //outputPorts[0]= dataOut[0] = tasa servidor 1;
    //outputPorts[1]= dataOut[1] = tasa servidor 2;

    // Set to true for helpful output.
    boolean debug = false;

    // Keep track of number of information agents.
    private int agentsSubscribed = 0;
    private int agentsLeftToRespond = 0;

    boolean firstRun = true;
    boolean dataUpdated = false;
    AID[] agentIDs = null;
    AID agentID = null;
    AID AgentCoordinatorID = null;

    TimeStepData tsd = new TimeStepData ();

    double colaS1=0;
    double colaS2=0;
```

```

double tasaS1=15.0;
double tasaS2=15.0;
double mediaS1=15.0;
double mediaS2=15.0;
double tiemposimulacion=0.0;
double mediafalla=15.0;
double fallaOperando=0;
double servidorSeleccionado=1;
double infoOtroHolon=0;
double tiempoSimFalla=400;
double tiempoRecuperacion=1200;
int fallaHabilitada=0;
int recuperacion=0;
int aux=1;
    /***VARIABLES PARA GENERAR NÚMEROS ALEATORIOS
float randomS1=0;
Random rndS1 = new Random();
float randomS2=0;
Random rndS2 = new Random();
//*****

public void setup() {

    // ****CREACIÓN Y DESPLIEGUE DE LA HMI DEL AGENTE****
myGui = new GuiHolonL1(this);
myGui.setVisible(true);
rndS1.setSeed(111);
rndS2.setSeed(112);
    addBehaviour(new filterBehaviour());

    /*
    * Register agent with directory facilitator of JADE along with the
    * services it provides (in this case, arithmetic and generic agent
services).
    * NB - To change properties after registering use the JADE
'modify' method.
    */

    String[] services = {"arithmetic", "Agent"};
    registerAgent(thisAgent, services, services);
//RegisterAgent(Agent a, String[] serviceName, String[] serviceType)

    /**@param a details of the agent being registered
    /**@param serviceName name of service that agent is registering with
    /**@param serviceType the type of service provided

    // Get agent number from argument provided by instantiator of this
agent.
    Object[] args = getArguments();
    if (args != null && args.length > 0) {
        String firstArg = args[0].toString();
        agentNumber = new Integer(firstArg).intValue();

    } else {

```

```

        // Terminate agent.
        System.out.println("No agent id");
        doDelete();
    }

    try {
        // Locate data provider.
        AgentCoordinatorID = getAgentIDsOfService(thisAgent,
            "InputsUpdate")[0];

        // Find agents that are interested in this agent's data.
        agentIDs = getAgentIDsOfService(thisAgent, "arithmetic");
//getAgentIDsOfService(Agent a, String serviceType)
//@param a details of agent making request
//@param serviceType
//@return Array of all the agent IDs offering this service
/*
 * N.B. If environment is dynamic would need to perform these
 * searches either more than once to check for new agents, or
 * implement some service subscription method to receive a
message
 * when new agents enter environment.
 */

    } catch (Exception e) {
        System.out.println("Error finding service: " + e);
    }

    agentsSubscribed = countSubscribingAgents(thisAgent, "arithmetic");

    if (debug)
        System.out.println("agentsSubscribed to arithmetic service:"
            + agentsSubscribed);

    agentsLeftToRespond = agentsSubscribed - 1;

}

class filterBehaviour extends CyclicBehaviour {
    public void action() {

        TimeStepData tsd = new TimeStepData();
        double[] dataArray;
        int length;

        // *****RECEPCIÓN DE MENSAJES
        ACLMessage msg = receive();

        if (msg != null) {
            String message = msg.getConversationId();

            //***** SE FILTRA EL MENSAJE POR LA ID DE LA CONVERSACIÓN
            //***** UPDATEDATA SIGNIFICA QUE VIENE DEL AGENTE COORDINADOR (ES DECIR,
            SIMULINK MATLAB)
            // If new data from AgentCoordinator (ie Simulink).
            if (message.equals("UpdateData")) {
                if (debug) {

```

```

        System.out.println("(Debug)Agent receiving new data
from Simulink. ");
        System.out.printf("\n");
    }

    try {
        tsd = (TimeStepData) msg.getContentObject();
    } catch (Exception e) {
        System.out.println("Exception with content object" + e);
    }

    /***Generación del siguiente número aleatorio***
    randomS1=rndS1.nextFloat();
    randomS2=rndS2.nextFloat();
    /**** Se extraen los datos del arreglo que llega desde simulink
    // Extract data from incoming array.
    length = tsd.getData().length;
    dataArray = new double[length];
    for (int i = 0; i < length; i++)
        dataArray[i] = tsd.getData()[i];

    if (debug) {
        System.out.println("(Debug)Incoming data: ");
        for (int j = 0; j < length; j++) {
            System.out.print(j + ": " + dataArray[j] + " ");
        }
        System.out.printf("\n");
    }

    /**** ASPECTOS PARTICULARES DE ESTE AGENTE
    /** Agent particular aspects begin here. *****/
    // ***** SE EXTRAEN LOS DATOS RELEVANTE DEL PUERTO DE INTERÉS
    // Extract the relevant data.
    colaS1 = (double) dataArray[inputPortsOfInterest[0]];
    colaS2 = (double) dataArray[inputPortsOfInterest[1]];
    tiemposimulacion = (double)
dataArray[inputPortsOfInterest[2]];
    if(tiemposimulacion<tiempoSimFalla)
    {

        tasaS1=-mediaS1*Math.log(1-randomS1);
        tasaS2=-mediaS2*Math.log(1-randomS2);
        myGui.campoEstadoServidores.setText(" OK ");
        fallaOperando=0;
    }
    else if(fallaHabilitada==0 &&
tiemposimulacion>=tiempoSimFalla)
    {
        tasaS1=-mediaS1*Math.log(1-randomS1);
        tasaS2=-mediaS2*Math.log(1-randomS2);
        myGui.campoEstadoServidores.setText(" OK ");
        fallaOperando=0;
    }
    /**** GENERACIÓN DE LA FALLA***

    else if(fallaHabilitada==1 &&
tiemposimulacion>=tiempoSimFalla && recuperacion==0)
    {
        fallaOperando=1;
        if(servidorSeleccionado==1)
        {

```

```

        mediaS1= mediafalla;
        mediaS2= mediafalla;
        tasaS1= -mediaS1*Math.log(1-randomS1);
        tasaS2= -mediaS2*Math.log(1-randomS2);

myGui.campoEstadoServidores.setText("Falla en Servidor 1");
    }

    else if(servidorSeleccionado==2)
    {
        mediaS1= mediafalla;
        mediaS2= mediafalla;
        tasaS1= -mediaS1*Math.log(1-randomS1);
        tasaS2= -mediaS2*Math.log(1-randomS2);

myGui.campoEstadoServidores.setText("Falla en Servidor 2");
    }

}

else if(fallaHabilitada==1 &&
tiemposimulacion>=tiempoSimFalla && recuperacion==1)
{
    fallaOperando=1;
    if(tiemposimulacion>=tiempoRecuperacion)
    {
        mediaS1=15.0;
        mediaS2=15.0;
        tasaS1= -mediaS1*Math.log(1-randomS1);
        tasaS2= -mediaS2*Math.log(1-randomS2);
myGui.campoEstadoServidores.setText("Falla Recuperada");
        fallaOperando=0;
    }
    else
    {
        if(servidorSeleccionado==1)
        {
            mediaS1= mediafalla;
            mediaS2= mediafalla;
            tasaS1= -mediaS1*Math.log(1-randomS1);
            tasaS2= -mediaS2*Math.log(1-randomS2);

myGui.campoEstadoServidores.setText("Falla en Servidor 1");
        }

        else if(servidorSeleccionado==2)
        {

            mediaS1= mediafalla;
            mediaS2= mediafalla;
            tasaS1= -mediaS1*Math.log(1-randomS1);
            tasaS2= -mediaS2*Math.log(1-randomS2);

myGui.campoEstadoServidores.setText("Falla en Servidor 2");
        }

    } }
}

```

```

        /*******MONITOREO DE PARÁMETROS ACTUALES*****
myGui.campoMediaTasaActualServ1.setText(Double.toString(mediaS1));
myGui.campoMediaTasaActualServ2.setText(Double.toString(mediaS2));

        /* Perform here any calculations required before using
data from
        * other agents.
        */

/**          Block          ends          here
*****/
    if (debug) {
        System.out.println(getLocalName() +
            " awaiting info from "
            + agentsLeftToRespond + " agent(s)");
    }
    dataUpdated = true;

    /*
    * Pass information about the new data to the
    * other agents.
    */
    if (agentsSubscribed > 1) {
        try {

/** Set information in dataStructure to be passed to other agents
*****/
            double [] dataToShare = {fallaOperando, colaS1, colaS2, recuperacion };

            int [] elementToChange = {0,1,2,3};
            TimeStepData tsdInner= new TimeStepData ();

            // Populate data structure with required data to exchange.
            tsdInner.setData(dataToShare);
            tsdInner.setElementsToChange(elementToChange);

/**          Block          ends          here
*****/

            // Send information off to other agents.
            for (int i = 0; i < agentIDs.length; i++) {
                if (!agentIDs[i].getLocalName().equals(
                    thisAgent.getLocalName())) {
                    sendObject(agentIDs[i], "agentDataL1",
                        tsdInner);
                }
            }
        } catch (Exception e) {
            System.out.println("Exception: " + e);
        }
    }

    if (agentsLeftToRespond == 0) {
        finishCalculationsThenSend();
    }

```

```

    }
}
//****RECEPCIÓN DE MENSAJES ENVIADOS POR OTROS
AGENTES*****
// Additional information from another agent.

//INFO DESDE HOLON RUTEO
if (message.equals("agentData")
    && !msg.getSender().getLocalName().equals(
        thisAgent.getLocalName())) {

    agentsLeftToRespond--;

    if (agentsLeftToRespond == 0 && dataUpdated) {
        finishCalculationsThenSend();
    }
} //*****
//INFO DESDE HOLON L0
if (message.equals("agentDataL0")
    && !msg.getSender().getLocalName().equals(
        thisAgent.getLocalName())) {
    agentsLeftToRespond--;

    if (agentsLeftToRespond == 0 && dataUpdated) {
        finishCalculationsThenSend();
    }
}
//INFO DESDE HOLON L2
if (message.equals("agentDataL2")
    && !msg.getSender().getLocalName().equals(
        thisAgent.getLocalName())) {

    agentsLeftToRespond--;

    if (agentsLeftToRespond == 0 && dataUpdated) {
        finishCalculationsThenSend();
    }
}
//INFO DESDE HOLON L3
if (message.equals("agentDataL3")
    && !msg.getSender().getLocalName().equals(
        thisAgent.getLocalName())) {
    agentsLeftToRespond--;

    if (agentsLeftToRespond == 0 && dataUpdated) {
        finishCalculationsThenSend();
    }
}
if (message.equals("DataAmended")) {
    if (debug)
System.out.println("Agent " + agentNumber + " finished.");

    replyToAgent(msg.getSender(), "ProcessingComplete");
}

if (message.equals("Shutting Down")) {
    takeDown(msg.getSender());
}

```

```

        }
    } else {
        block();
    }
}

/**
 * Finalise calculations and return to AgentCoordinator and Simulink.
 */
public void finishCalculationsThenSend() {

/**
    Agent specific task inside this block
    *****/
    // Carry out any final calculation before returning back to Simulink
and
    // specified port.

    //***** ENVÍO DE DATO A SIMULINK*****

    double[] dataOut = new double[outputPorts.length];

    dataOut[0] = tasaS1;
    dataOut[1] = tasaS2;

/**
    Block ends here
    *****/

    tsd.setData(dataOut);
    tsd.setElementsToChange(outputPorts);

    // Send modified data back to Simulink.
    ACLMessage reply = new ACLMessage(ACLMessage.INFORM);
    reply.setConversationId("ProcessedData");

    try {
        reply.setContentObject(tsd);
    } catch (Exception e) {
        System.out.println("Exception setting tsd: " + e);
    }

    if (debug)
        System.out.println(getLocalName() + " finished, sending data to
"
                                + AgentCoordinatorID.getLocalName());
    reply.addReceiver(AgentCoordinatorID);
    send(reply);
    agentsLeftToRespond = agentsSubscribed - 1;
    dataUpdated = false;
}

```

```
/**
 * Terminate agent.
 *
 * @param senderAID AID
 */
protected void takeDown(AID senderAID) {
    try {
        System.out.println("Deregistering and closing " + thisAgent);
        DFService.deregister(thisAgent);
    } catch (FIPAException fe) {
        System.out.println("Problem deregistering " + fe);
    }
    if (senderAID != null)
        contactAgent(senderAID, "Agent Deregistered");

    doDelete();
}
}
```



## ANEXO 3

### Gráficos de Resultados

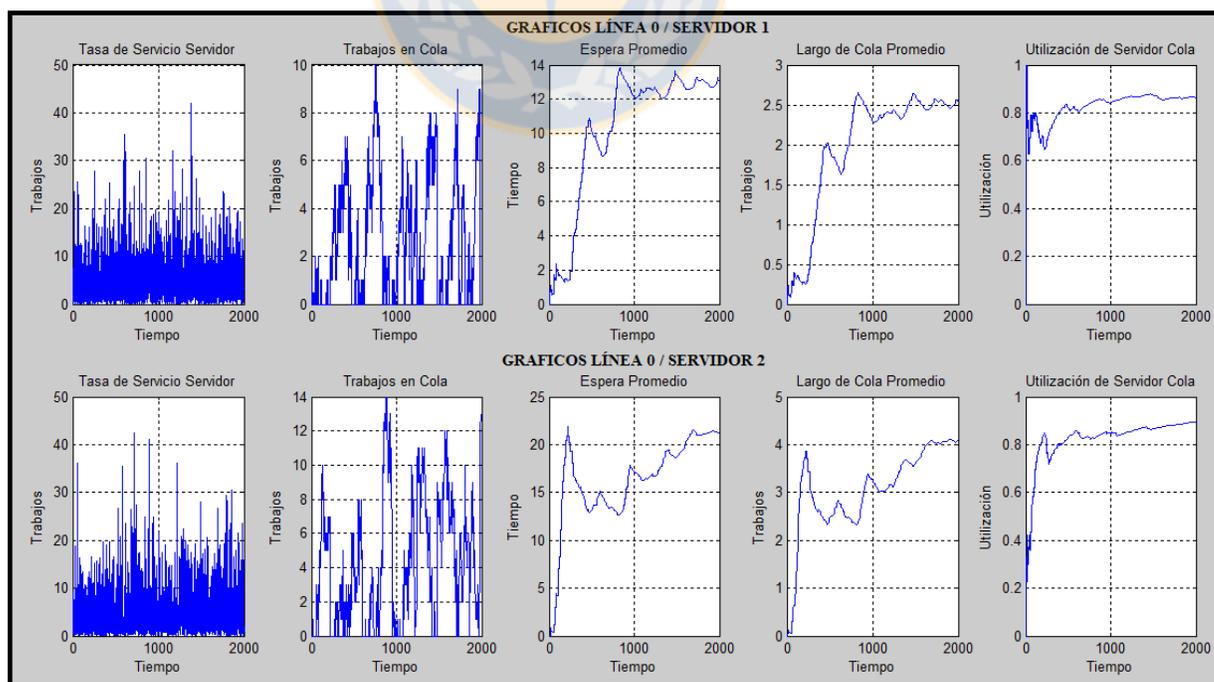
En el presente anexo se muestra en detalle los gráficos resultantes de los escenarios planteados en el capítulo 3.

El color de las gráficas de definen a continuación:

- Línea Azul ( — ) : Grafica los resultados obtenidos con el sistema de control convencional y sin falla (Escenario Base).
- Línea Roja ( - - - ) : Grafica los resultados obtenidos con el sistema convencional con la falla del escenario correspondiente.
- Línea Verde ( - · - · - ) : Grafica los resultados obtenidos con el sistema CRC-H y con la falla del escenario correspondiente con y sin recuperación.

#### 3.3.4.10 A.3.1 Escenario base: Control Convencional sin Falla

En la Figura 50 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{01}$  y  $S_{02}$ , respectivamente.



**Figura 50: Gráficos Línea 0 / Servidores 1 y 2, Control Convencional sin falla**

En la Figura 51 se muestran las gráficas asociadas a tasa de servicio, trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.

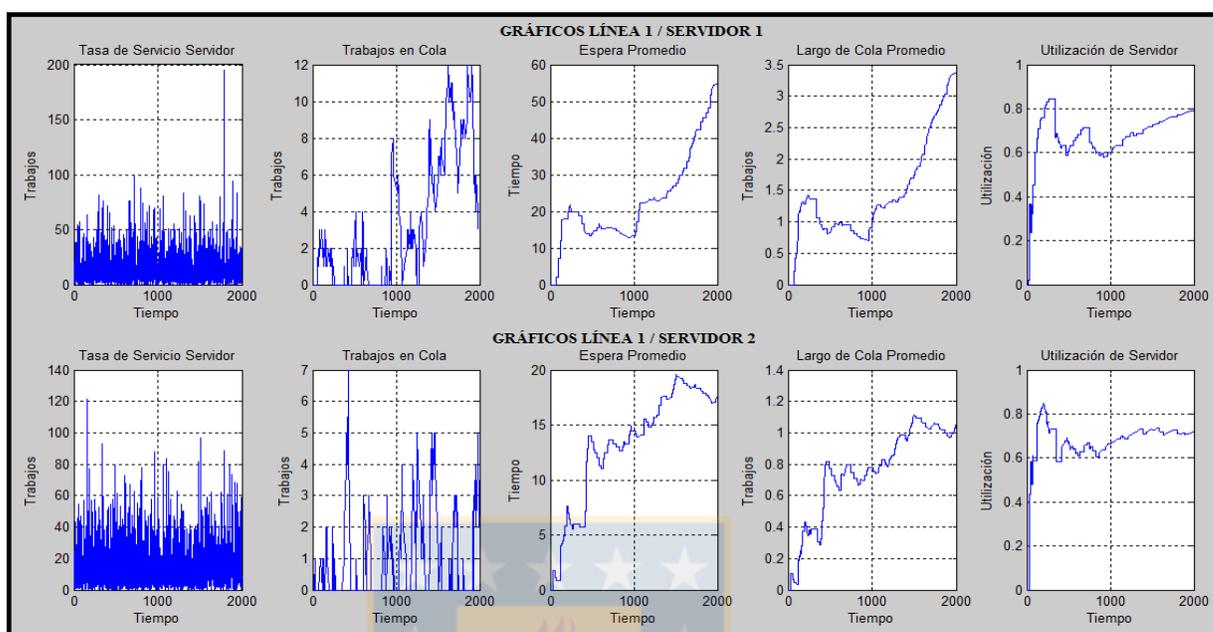


Figura 51: Gráficos Línea 1 / Servidores 1 y 2, Control Convencional sin falla

En la Figura 52 se muestran las gráficas asociadas a tasa de servicio, trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.

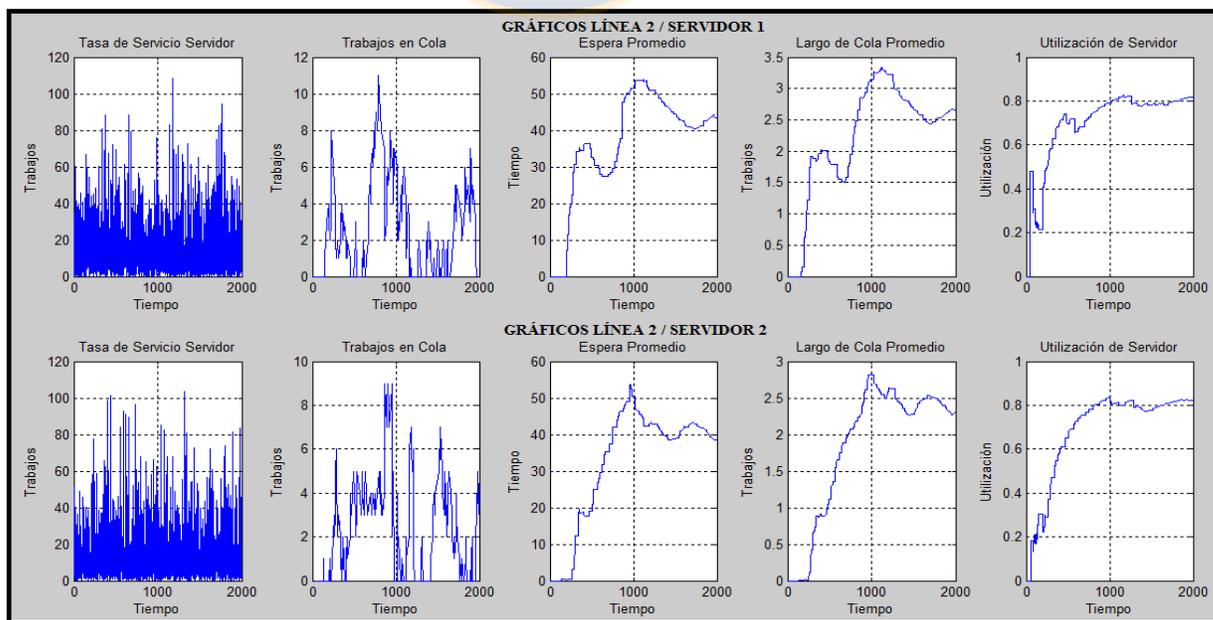


Figura 52: Gráficos Línea 2 / Servidores 1 y 2, Control Convencional sin falla

En la Figura 53 se muestran las gráficas asociadas a tasa de servicio, trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.

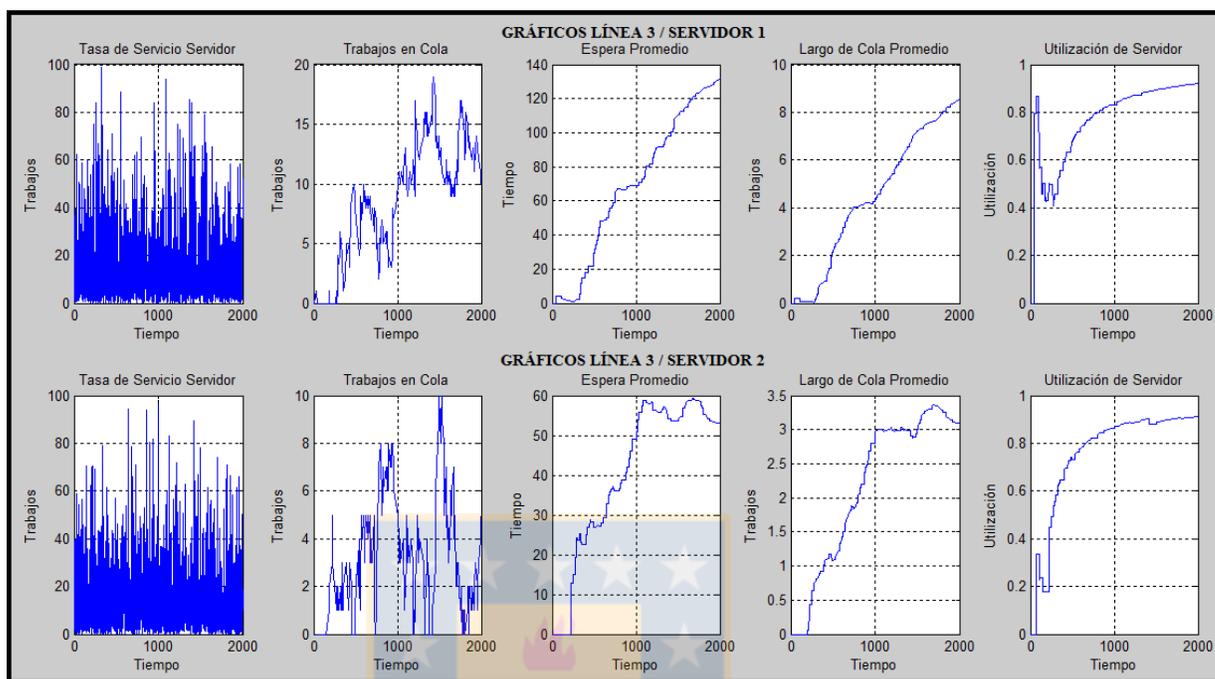


Figura 53: Gráficos Línea 3 / Servidores 1 y 2, Control Convencional sin falla

En la Figura 54 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .

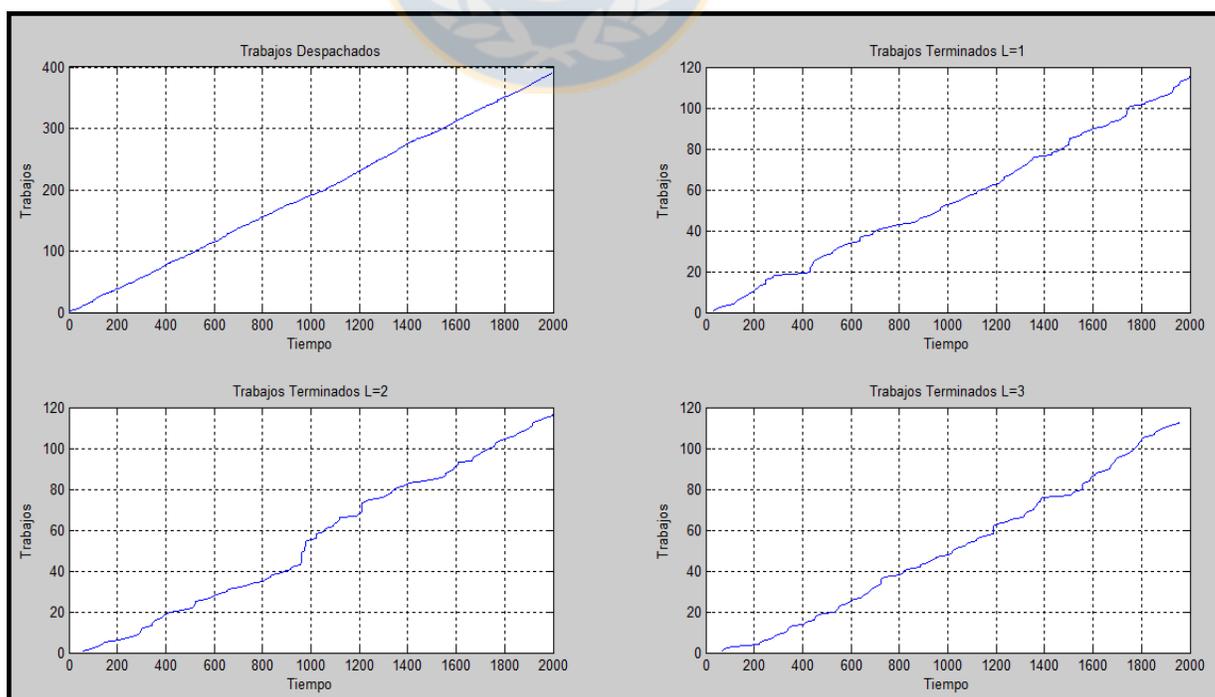


Figura 54: Trabajos despachados y terminados en  $L1$ ,  $L2$  y  $L3$ , Control Convencional sin falla

### 3.3.4.11 A.3.2 Escenario 1: Falla en el servidor S11, sin recuperación

En la Figura 55 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.

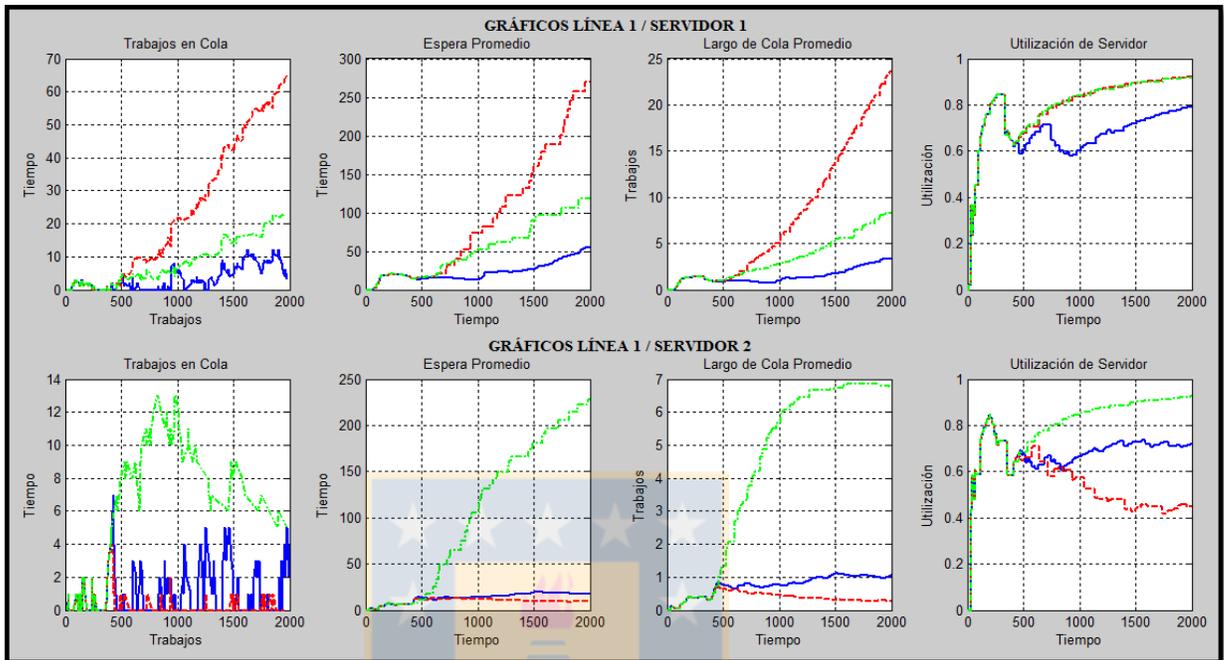


Figura 55: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 1  
 (—) Escenario Base (---) Ctrl Convencional con falla (····) CRC-H con falla

En la Figura 56 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.

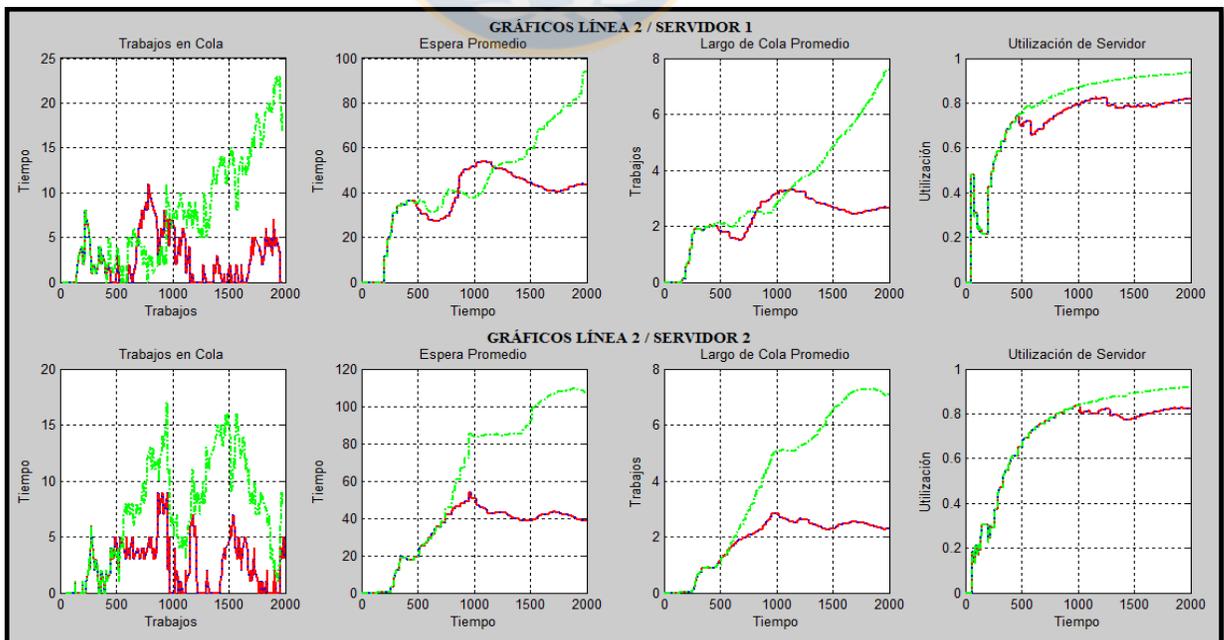
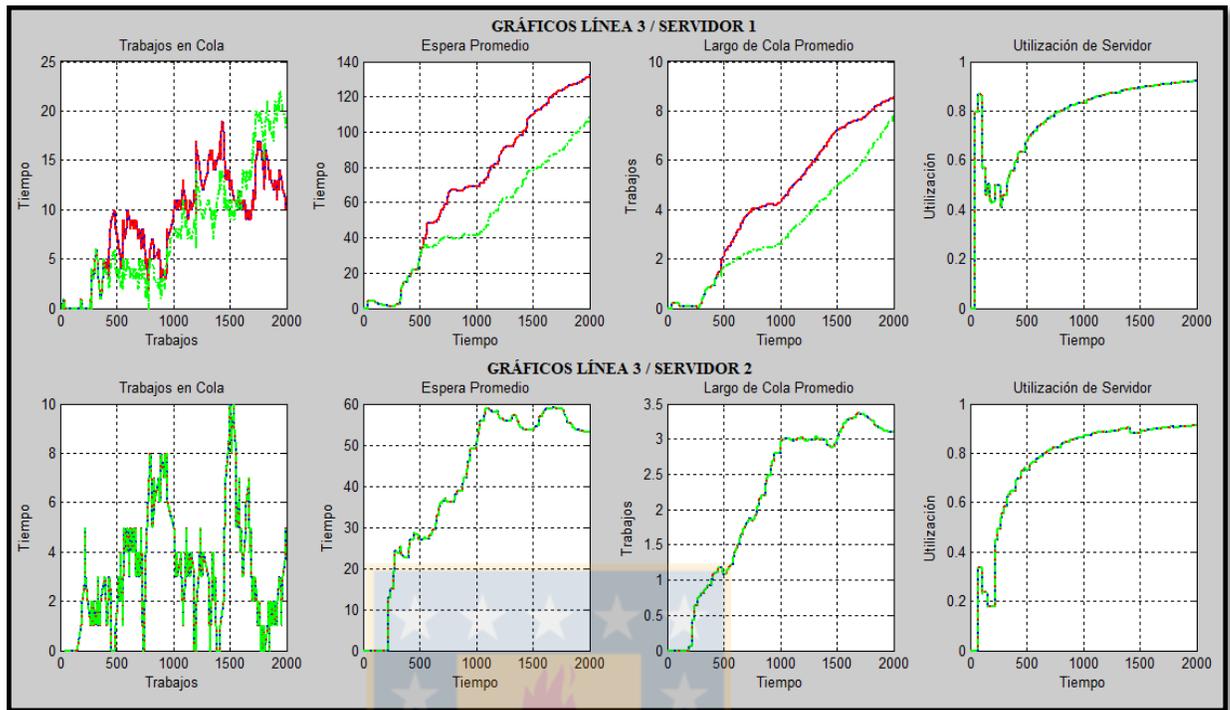


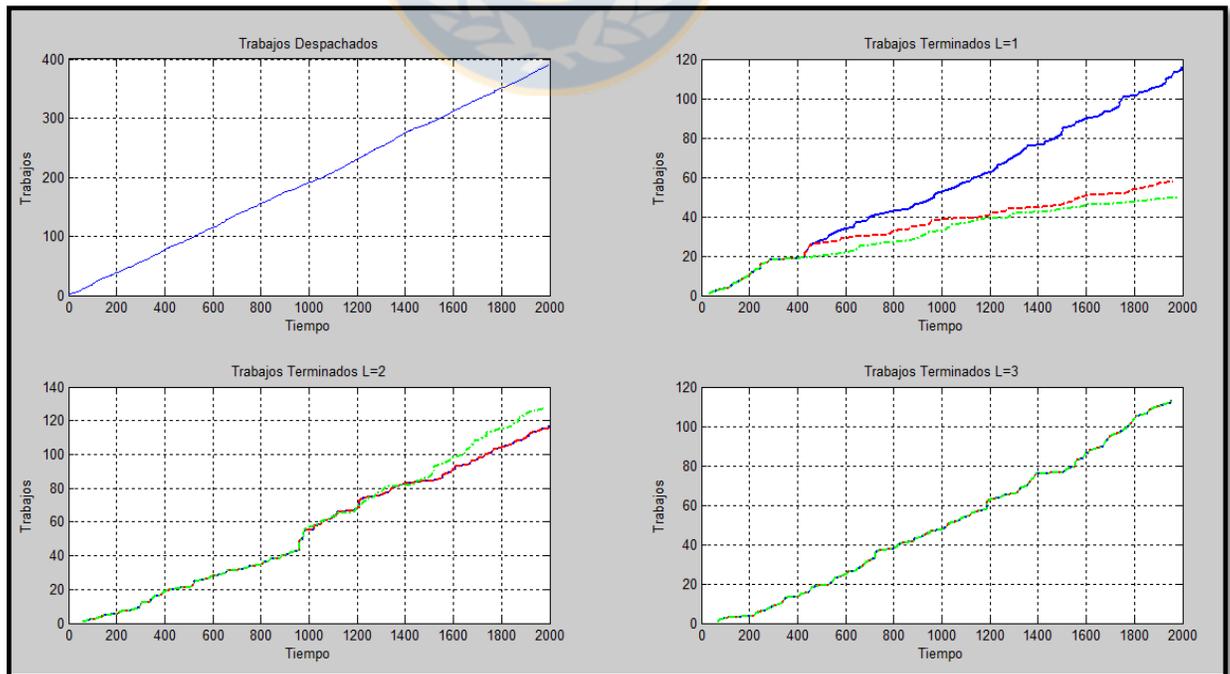
Figura 56: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 1  
 (—) Escenario Base (---) Ctrl Convencional con falla (····) CRC-H con falla

En la Figura 57 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 57: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 1**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

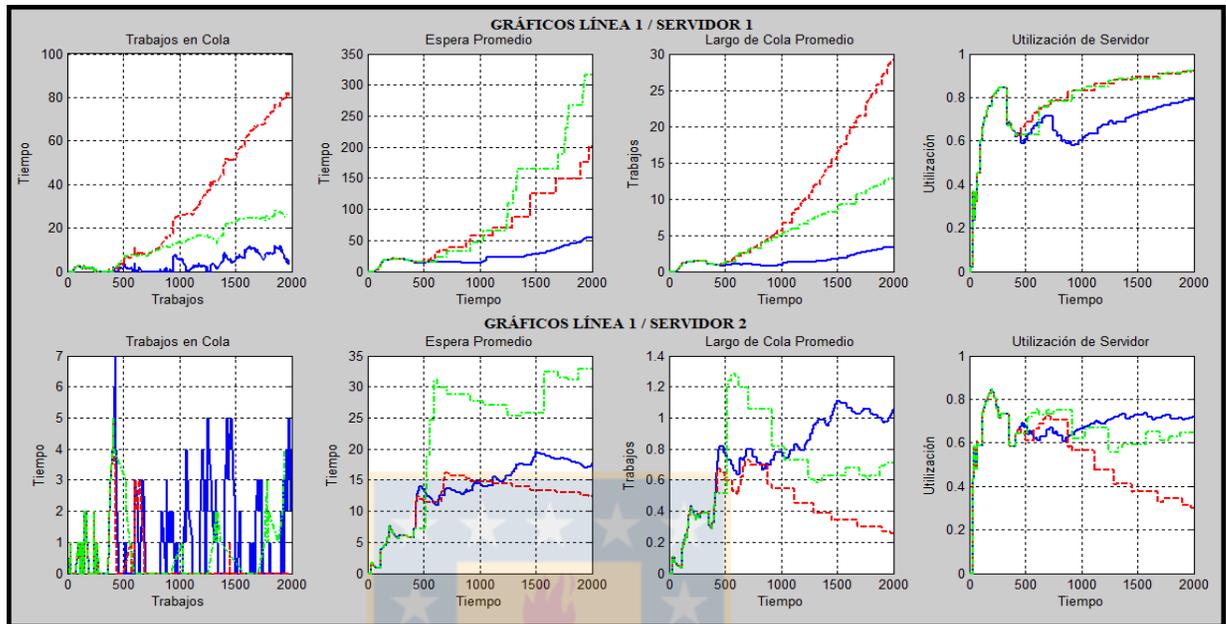
En la Figura 58 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 58: Trabajos despachados y Terminados en L1, L2 y L3, Escenario 1**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

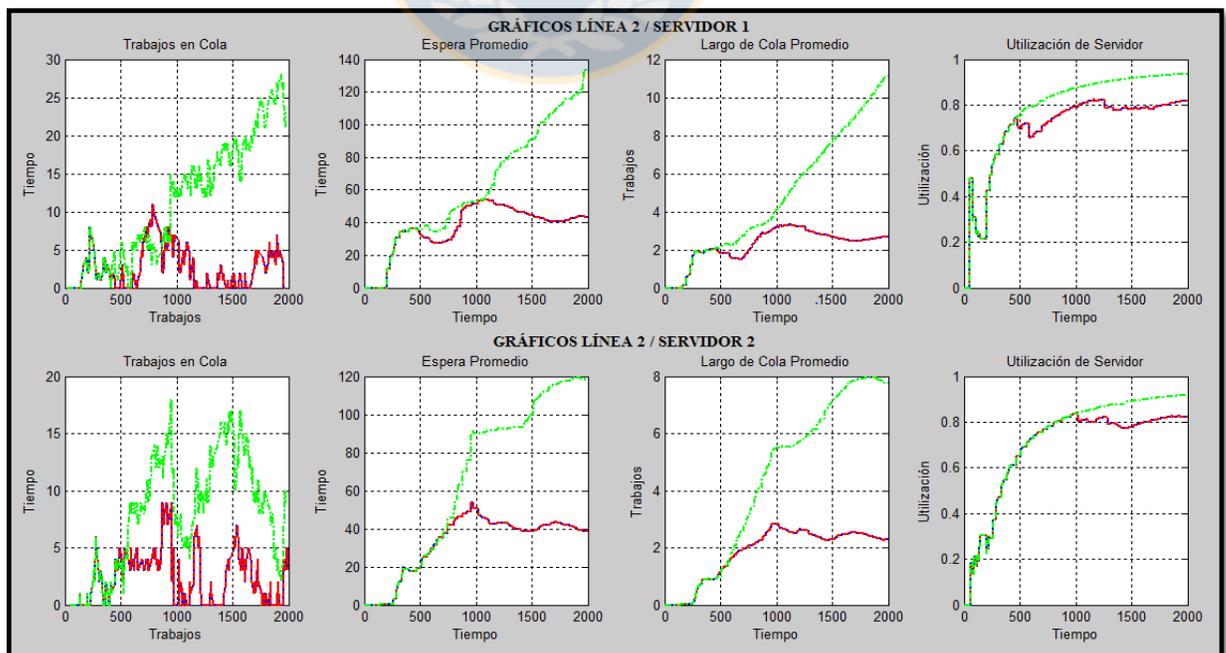
### 3.3.4.12 A.3.3 Escenario 2: Falla en el servidor S11, sin recuperación

En la Figura 60 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.



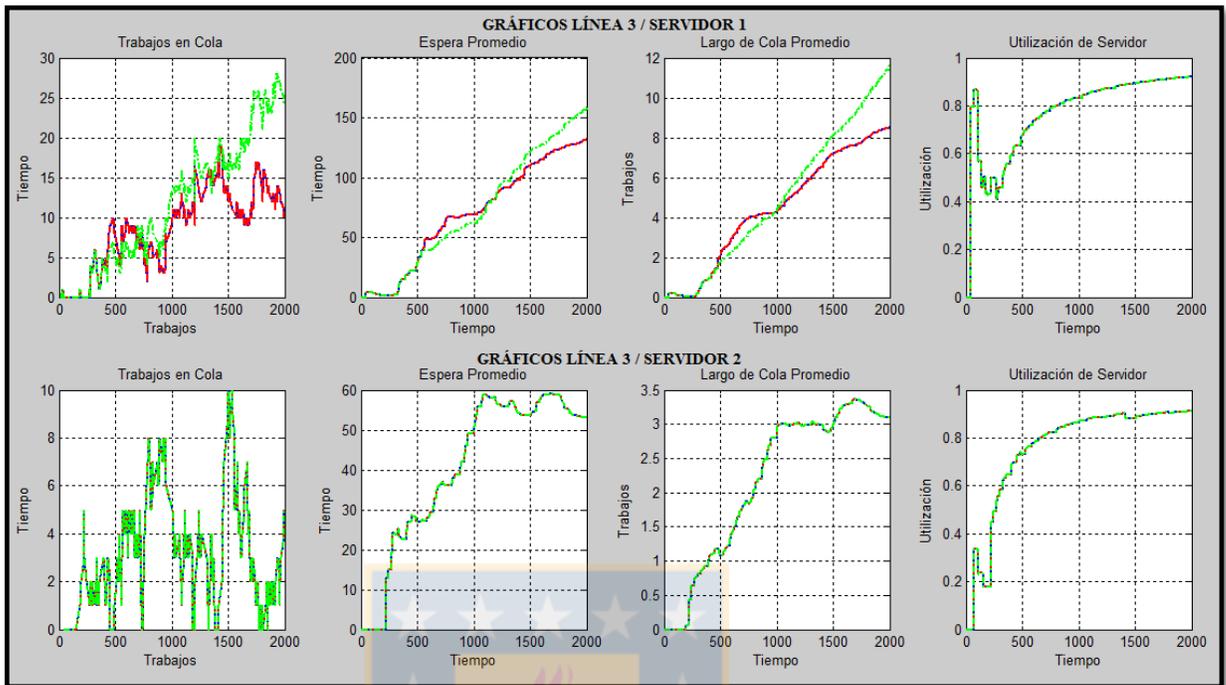
**Figura 59: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 2**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 59 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.



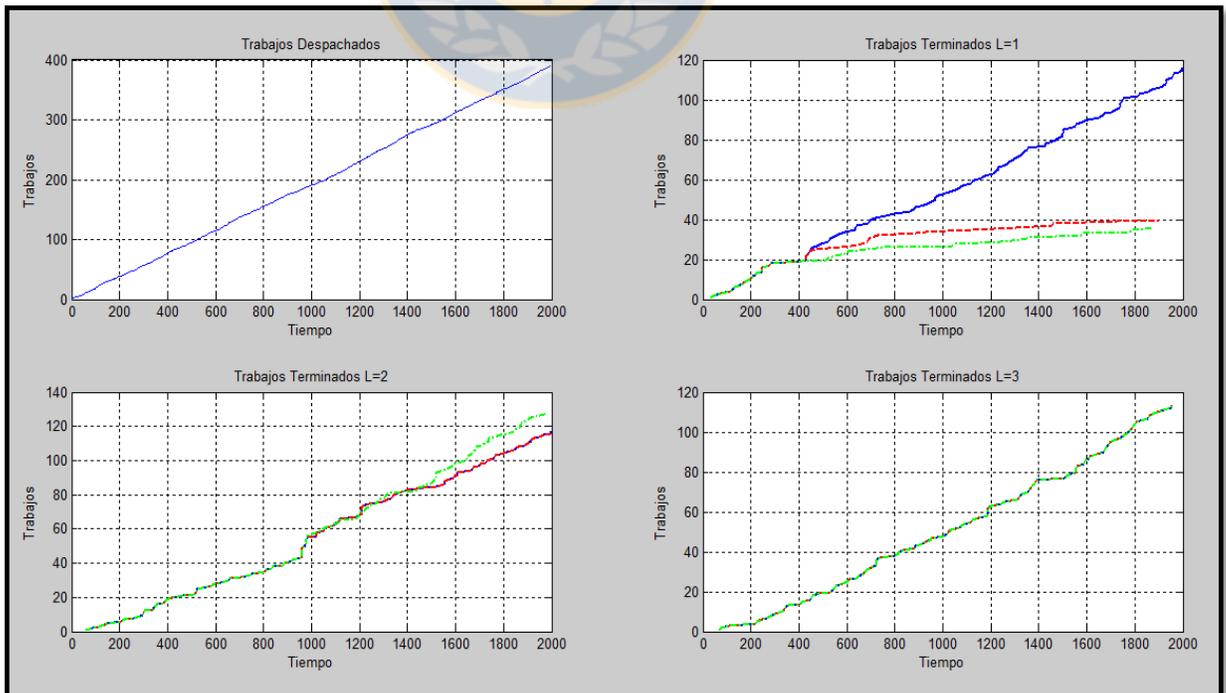
**Figura 60: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 2**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 61 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 61: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 2**  
 (—) Escenario Base ( - - - ) Ctrl Convencional con falla ( - · - · ) CRC-H con falla

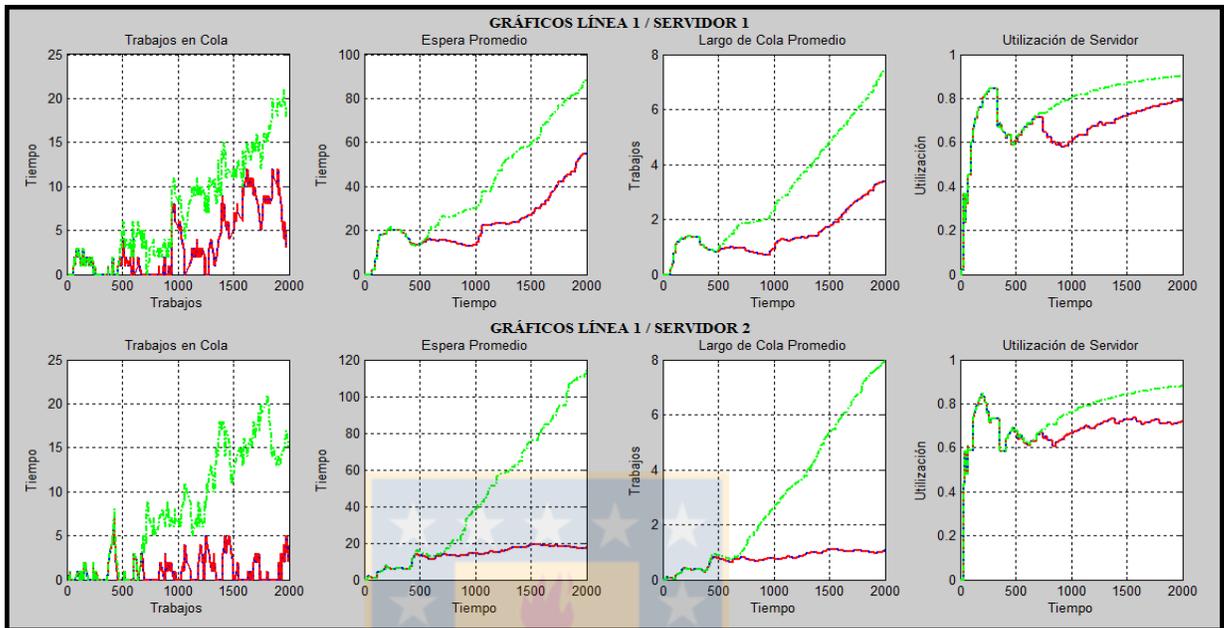
En la Figura 62 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 62: Trabajos despachados y terminados en L1, L2 y L3, Escenario 2**  
 (—) Escenario Base ( - - - ) Ctrl Convencional con falla ( - · - · ) CRC-H con falla

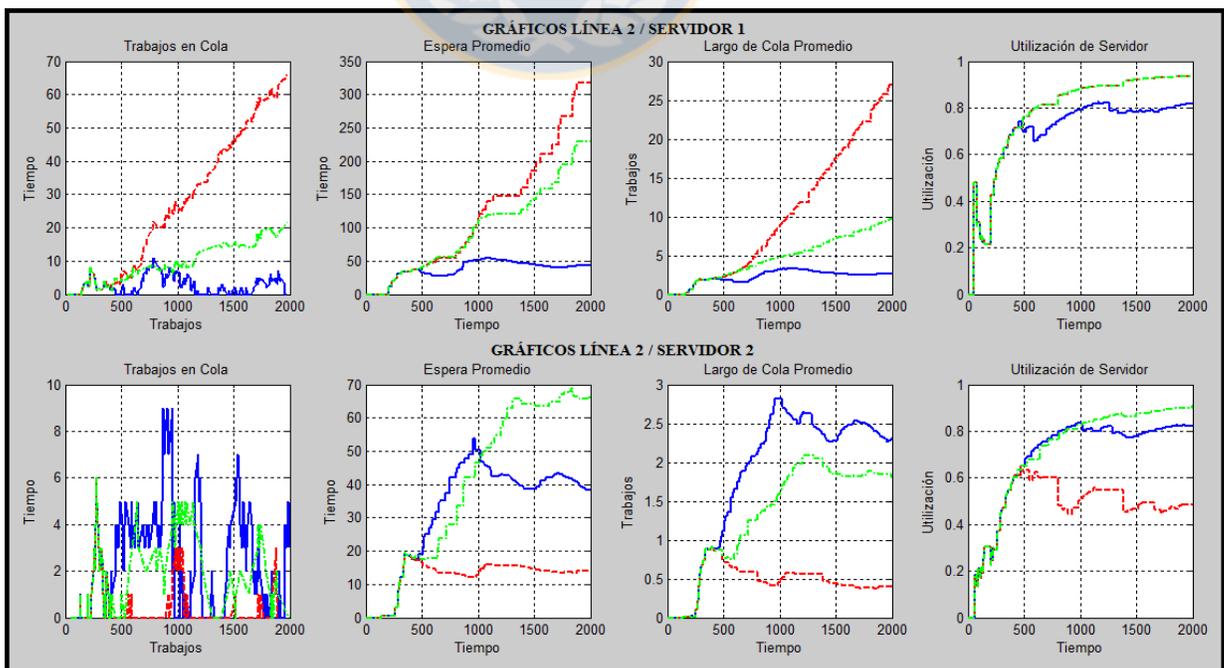
### 3.3.4.13 A.3.4 Escenario 3: Falla en el servidor $S_{21}$ , sin recuperación

En la Figura 63 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.



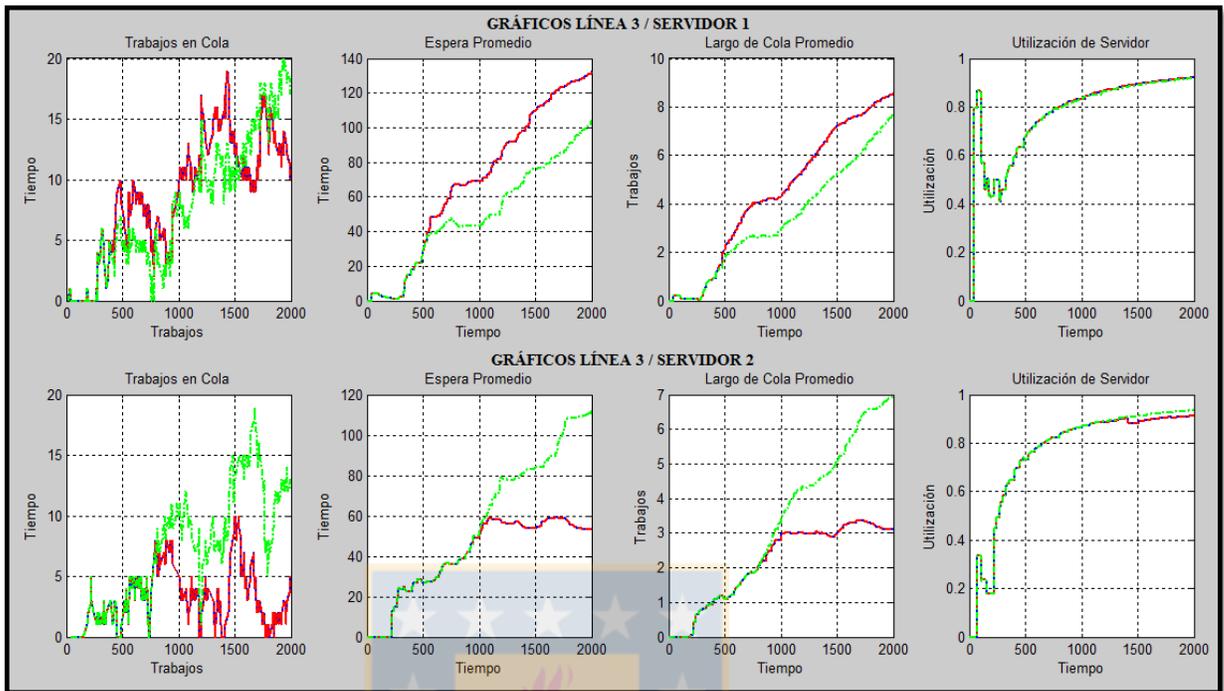
**Figura 63: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 3**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.-) CRC-H con falla

En la Figura 64 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.



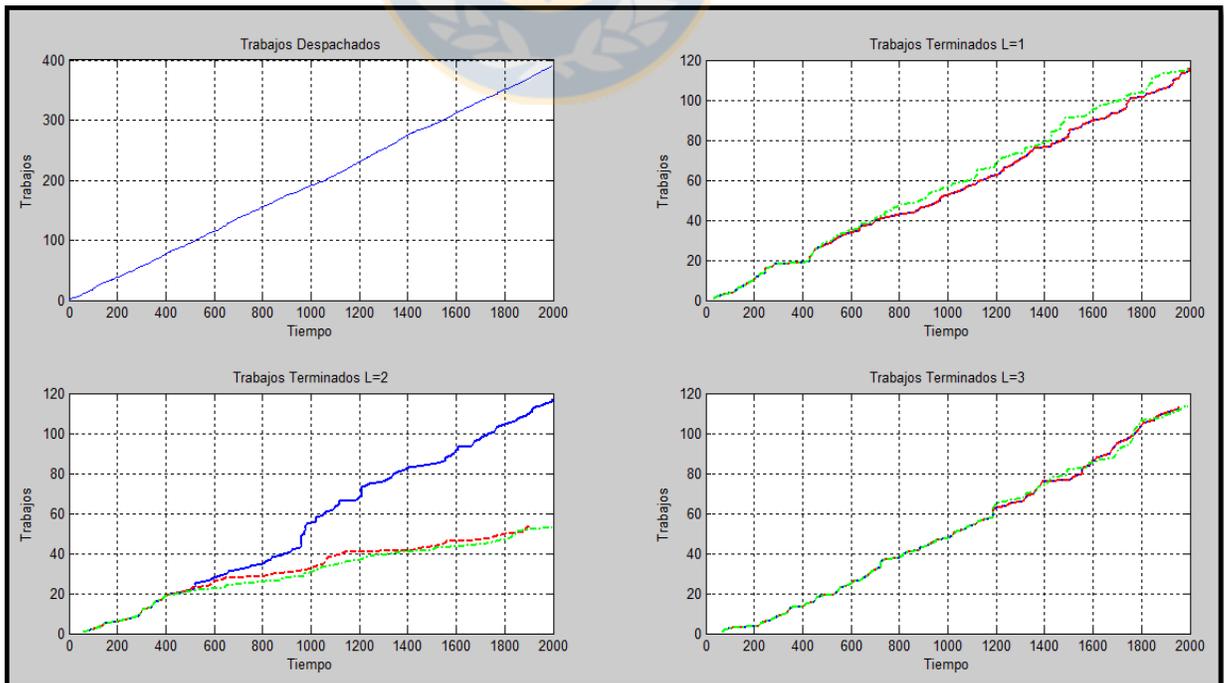
**Figura 64: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 3**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.-) CRC-H con falla

En la Figura 65 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 65: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 3**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

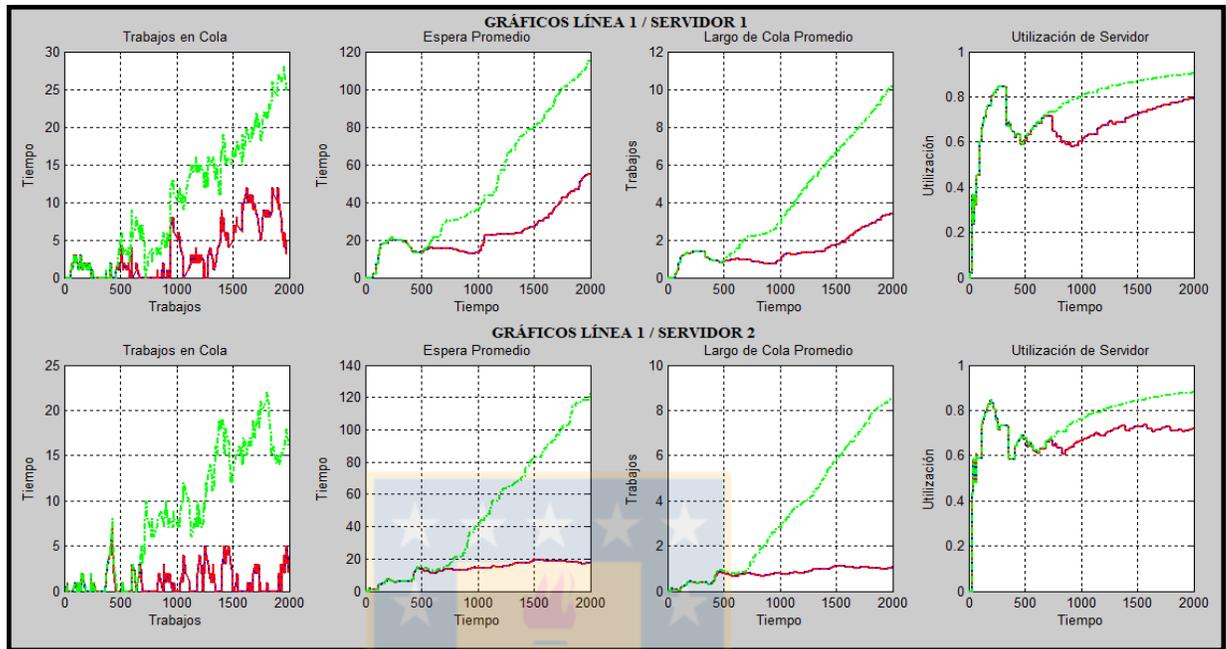
En la Figura 66 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 66: Trabajos despachados y terminados en L1, L2 y L3, Escenario 3**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

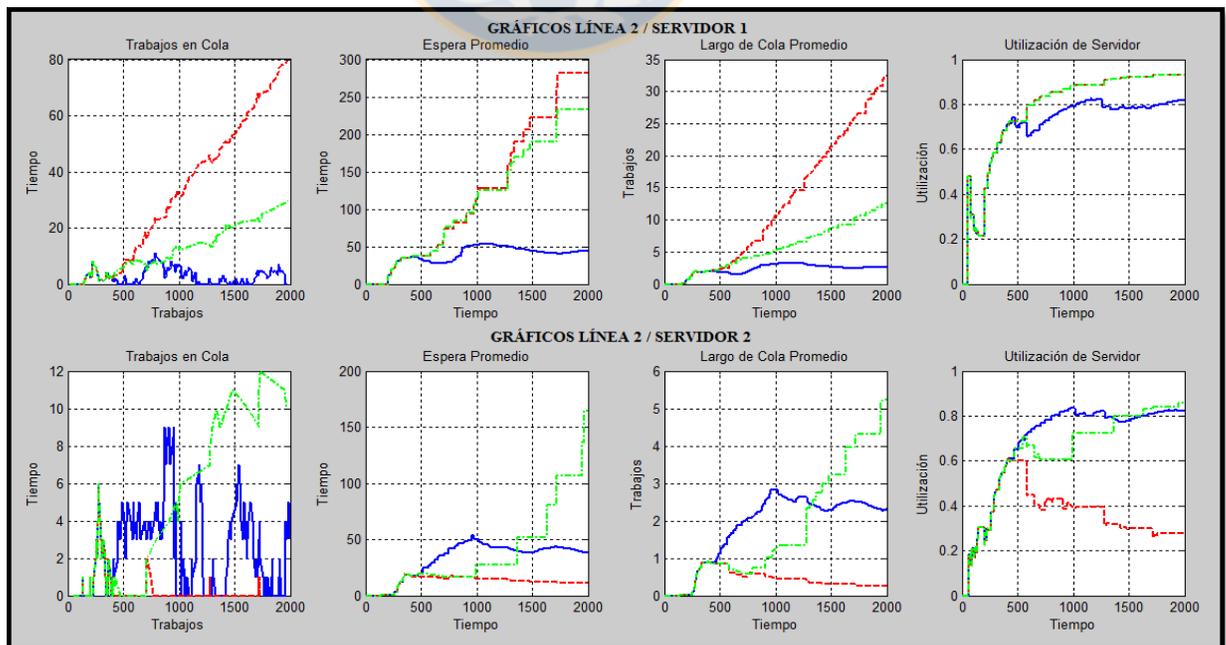
### 3.3.4.14 A.3.5 Escenario 4: Falla en el servidor $S_{21}$ , sin recuperación

En la Figura 67 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.



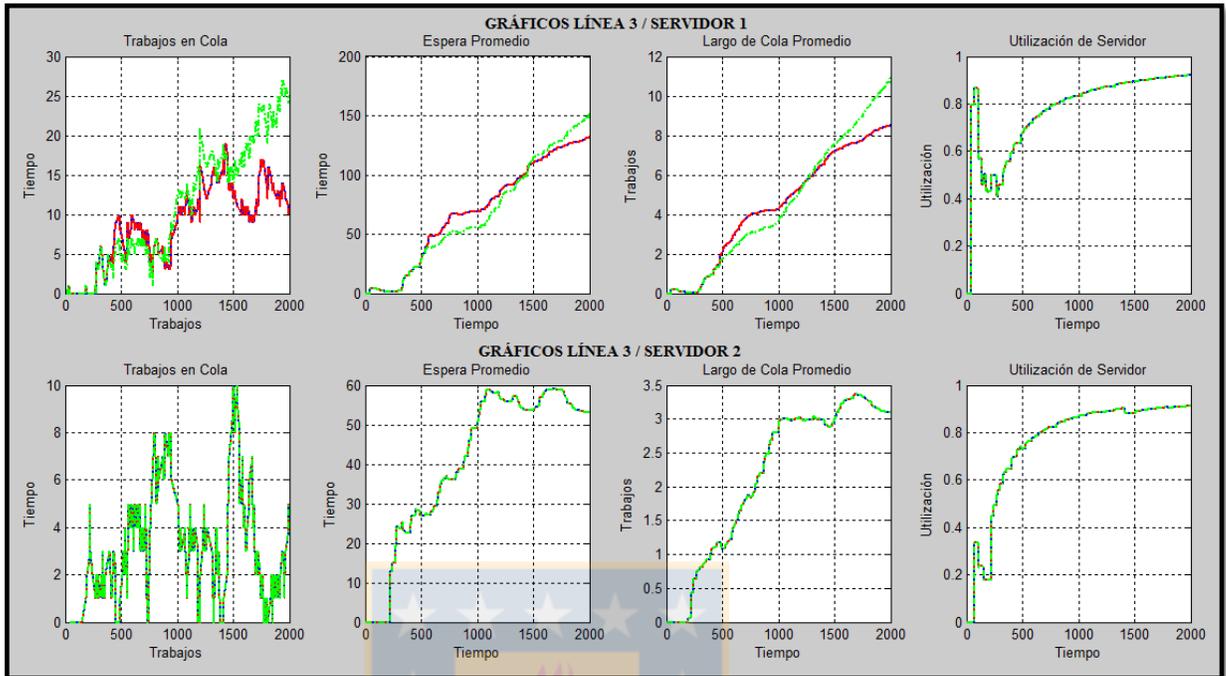
**Figura 67: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 4**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

En la Figura 68 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.



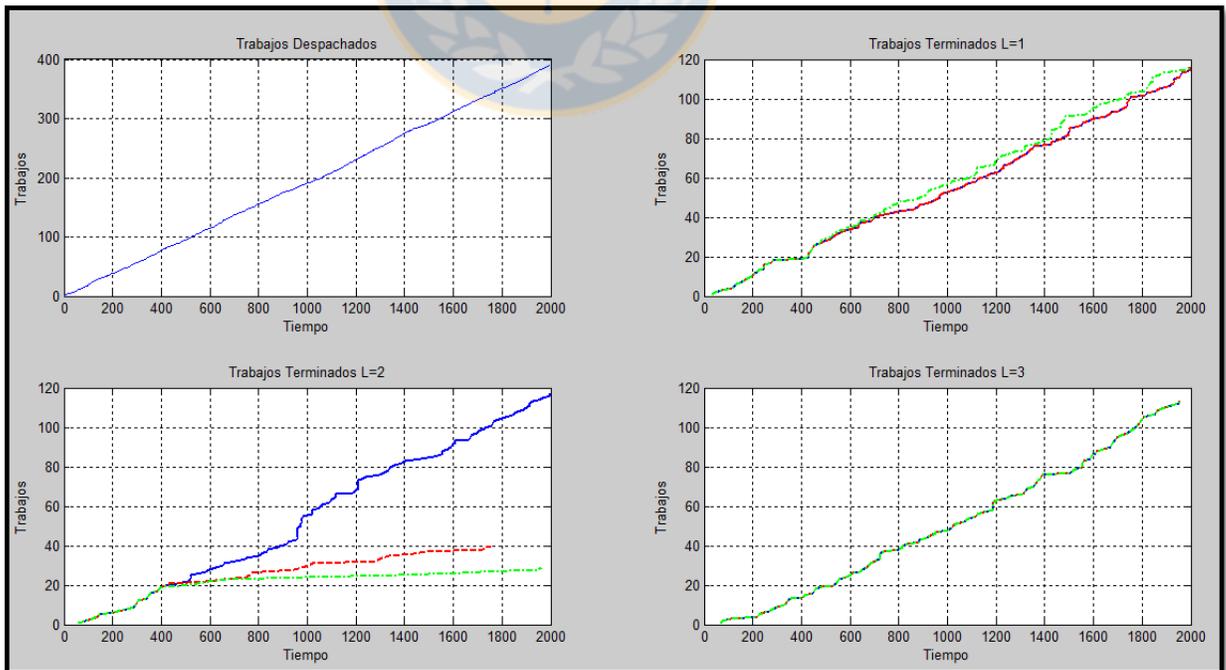
**Figura 68: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 4**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

En la Figura 69 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 69: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 4**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

En la Figura 70 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 70: Trabajos despachados y terminados en L1, L2 y L3, Escenario 4**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

### 3.3.4.15 A.3.6 Escenario 5: Falla en el servidor $S_{31}$ , sin recuperación

En la Figura 71 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.

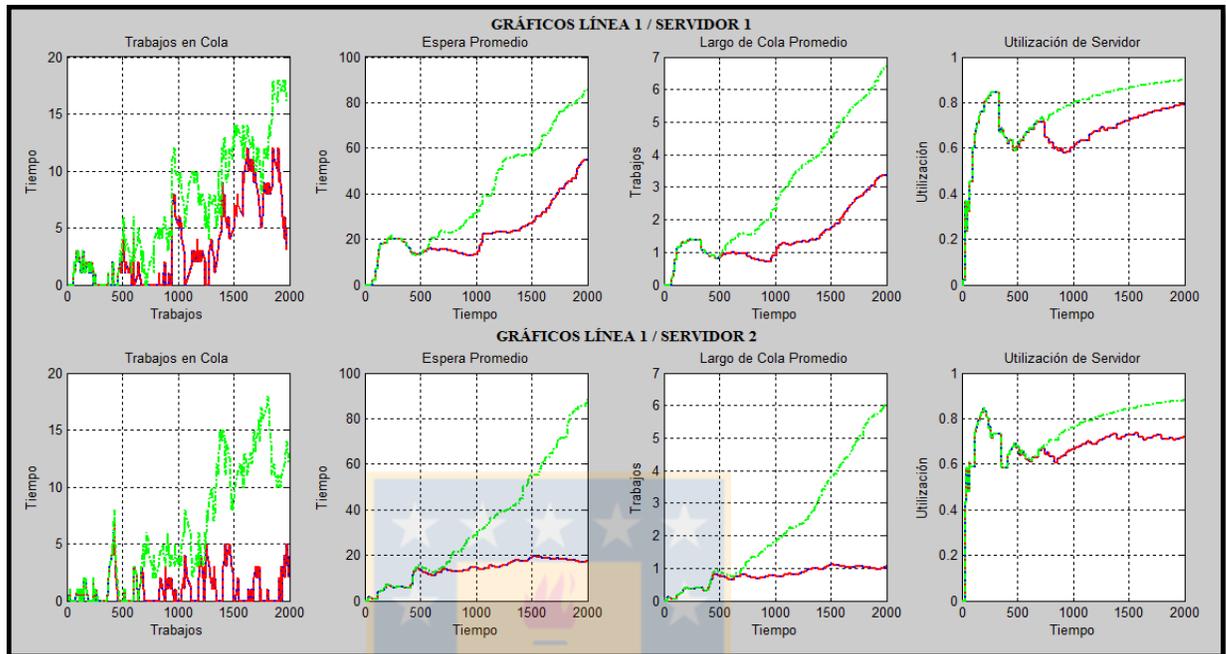


Figura 71: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 5  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 72 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.

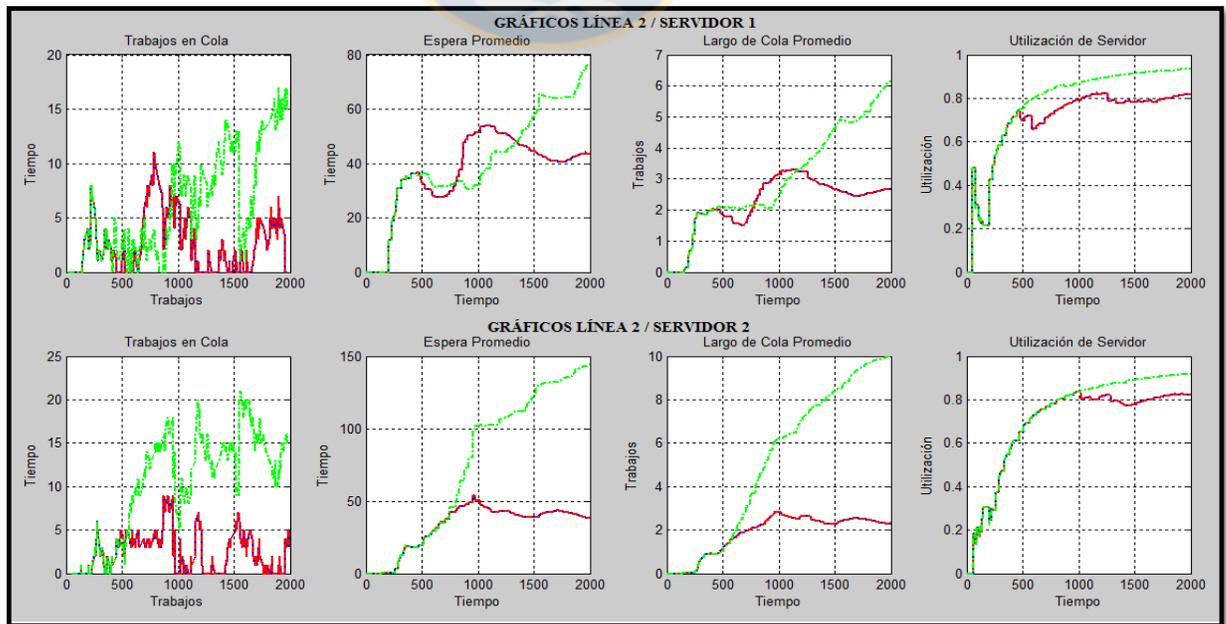
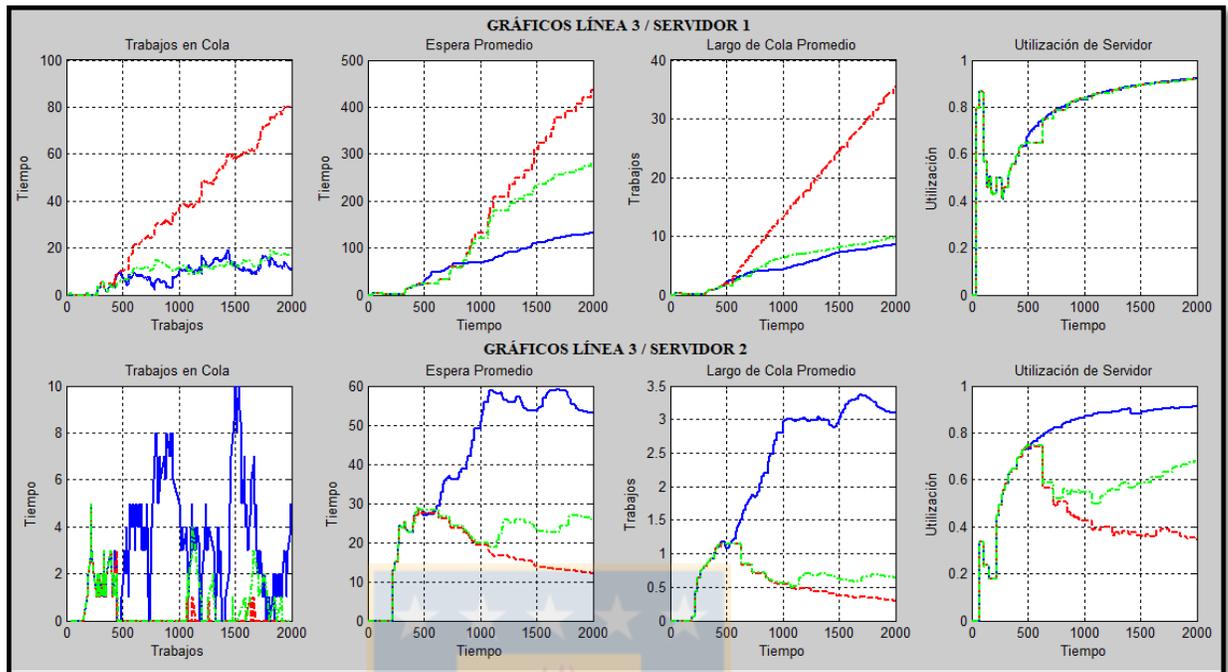


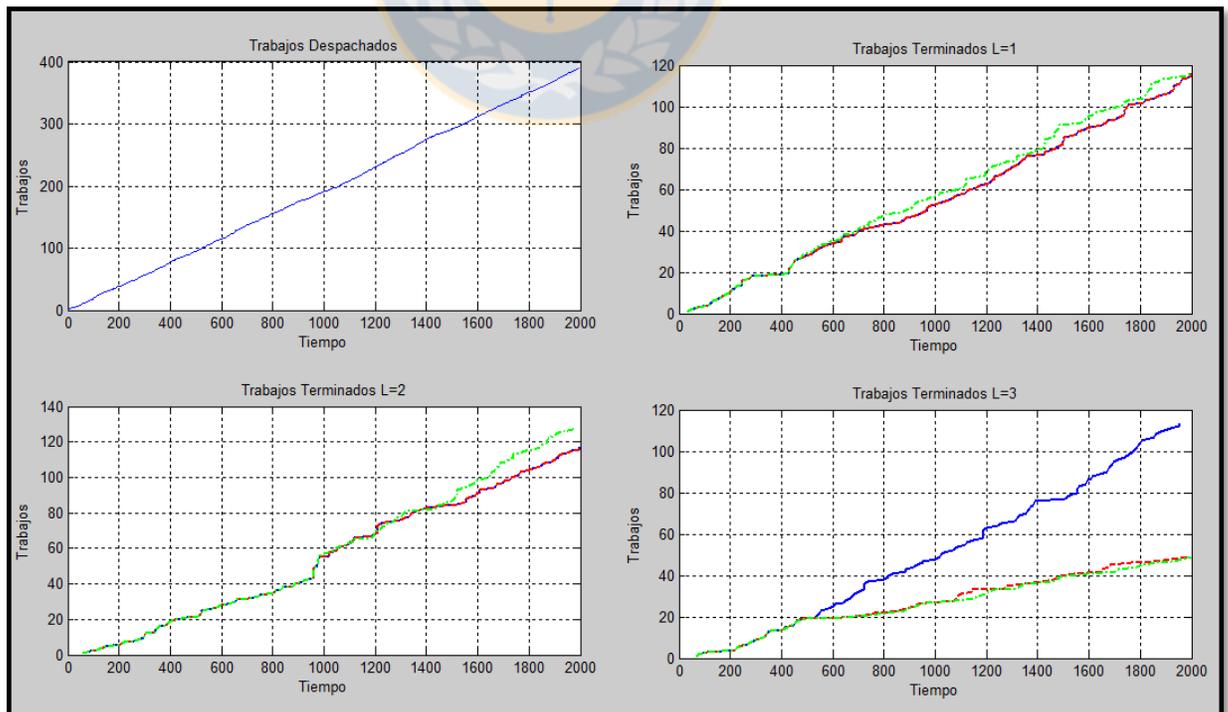
Figura 72: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 5  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 73 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 73: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 5**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

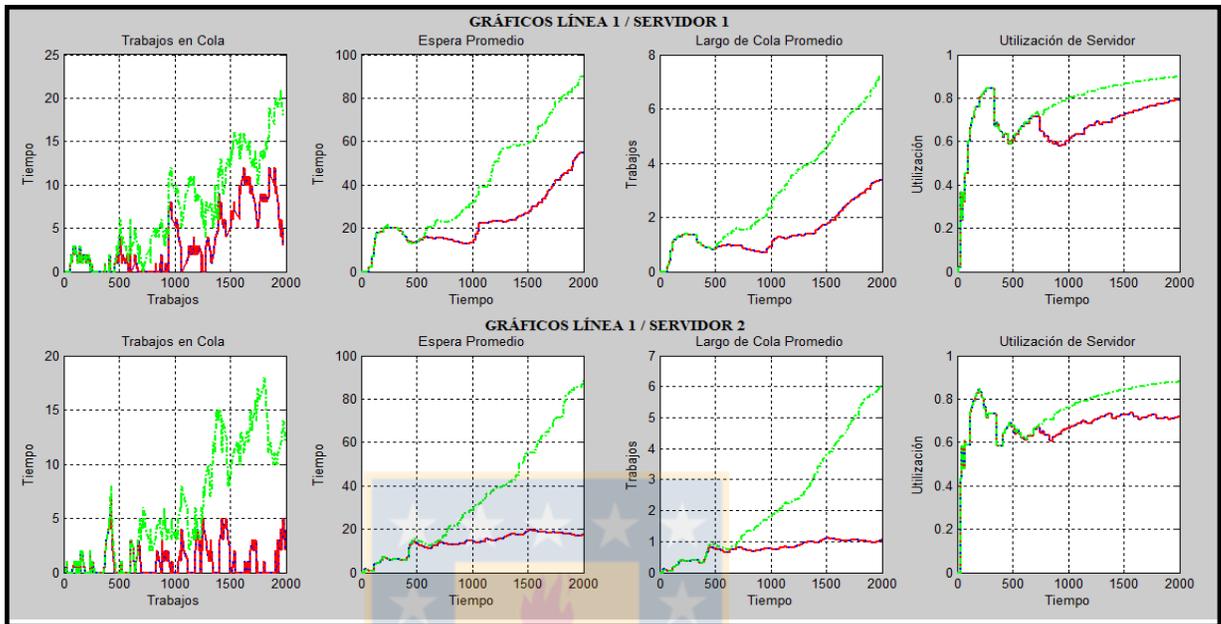
En la Figura 74 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 74: Trabajos despachados y terminados en L1, L2 y L3, Escenario 5**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

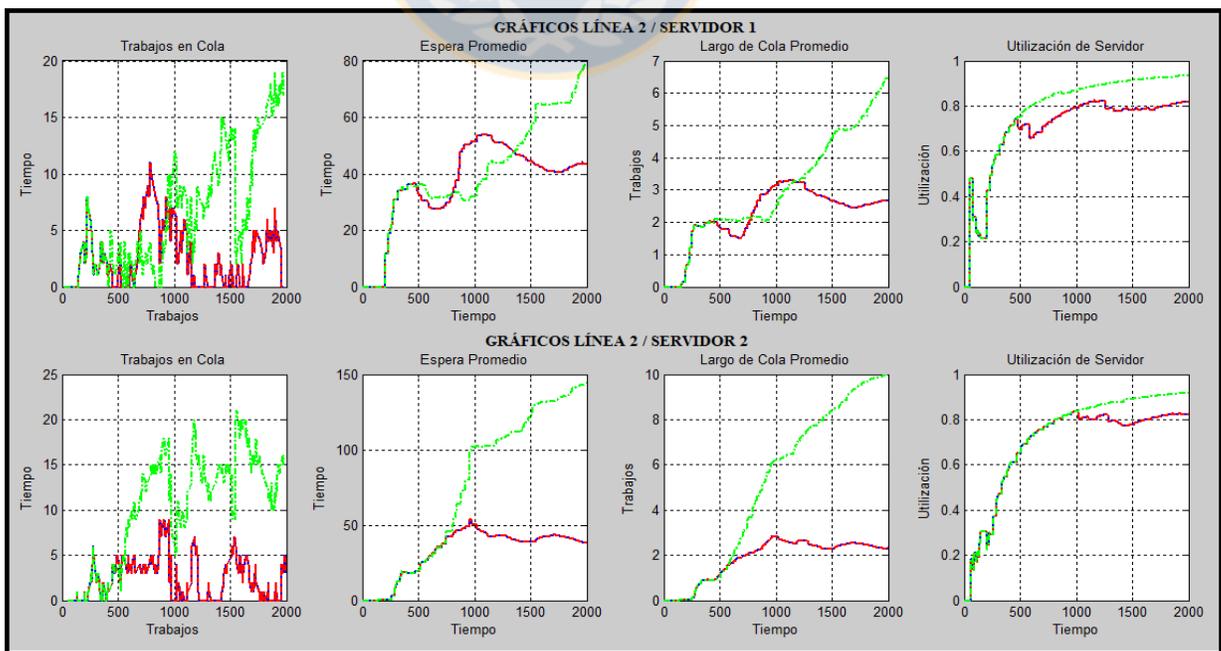
### 3.3.4.16 A.3.7 Escenario 6: Falla en el servidor $S_{31}$ , sin recuperación

En la Figura 75 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.



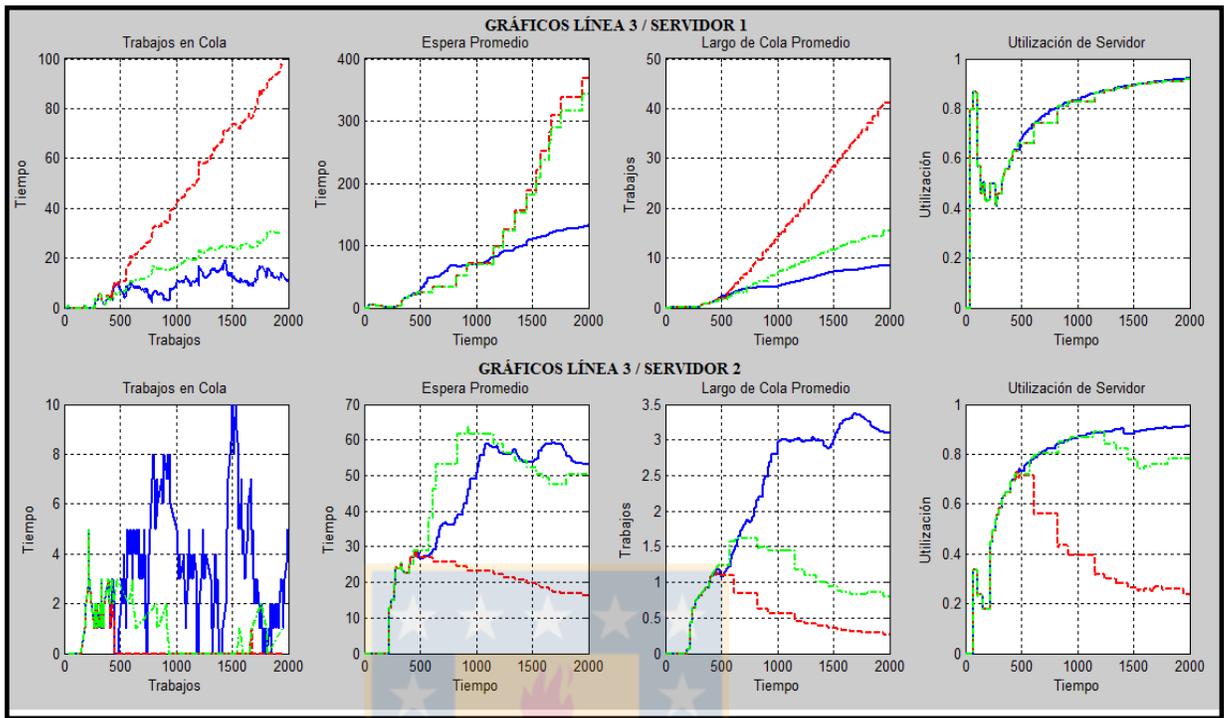
**Figura 75: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 6**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 76 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.



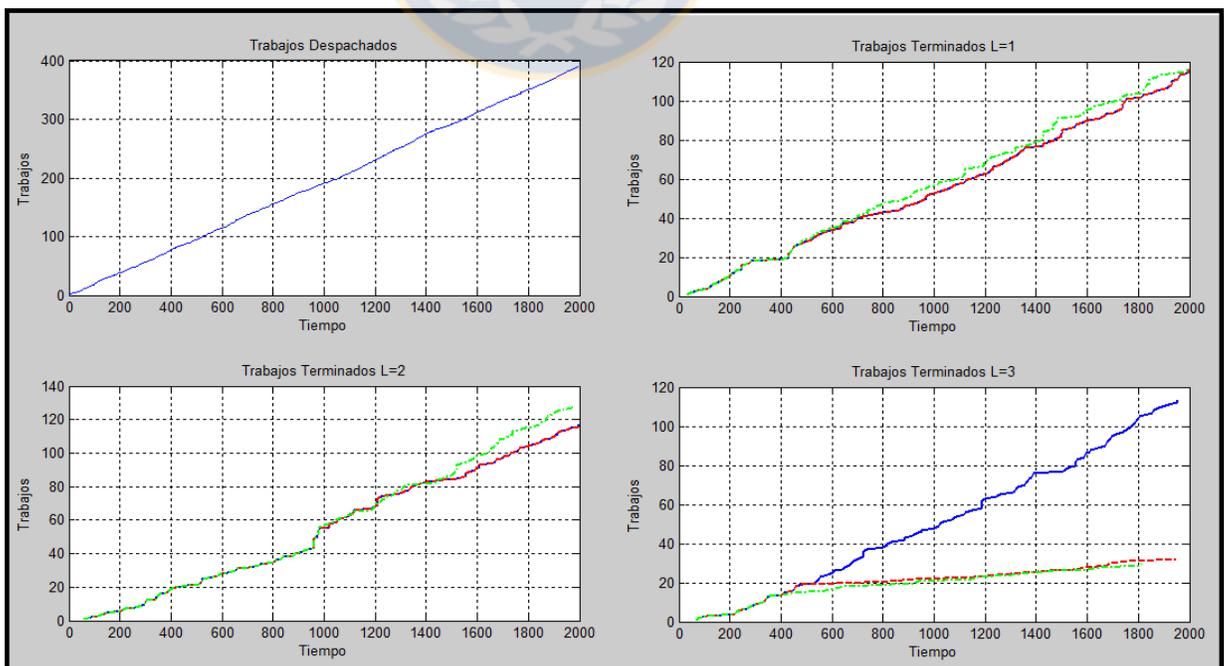
**Figura 76: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 6**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 77 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



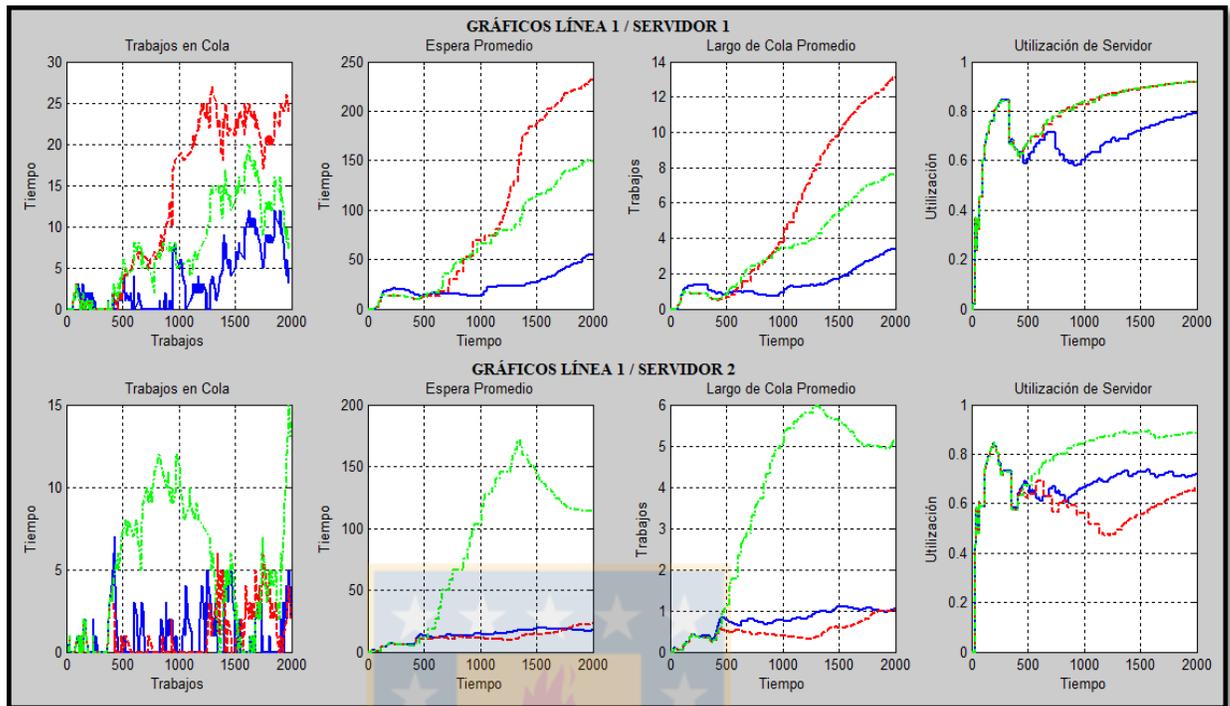
**Figura 77: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 6**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 78 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



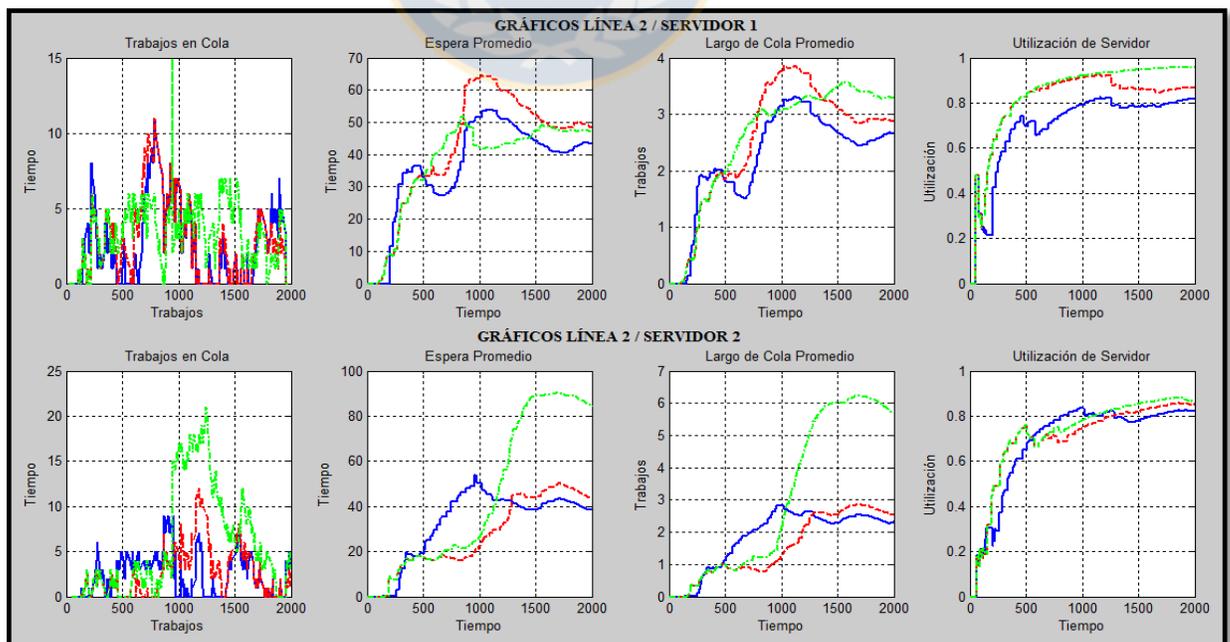
**Figura 78: Trabajos despachados y terminados en L1, L2 y L3, Escenario 6**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 79 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.



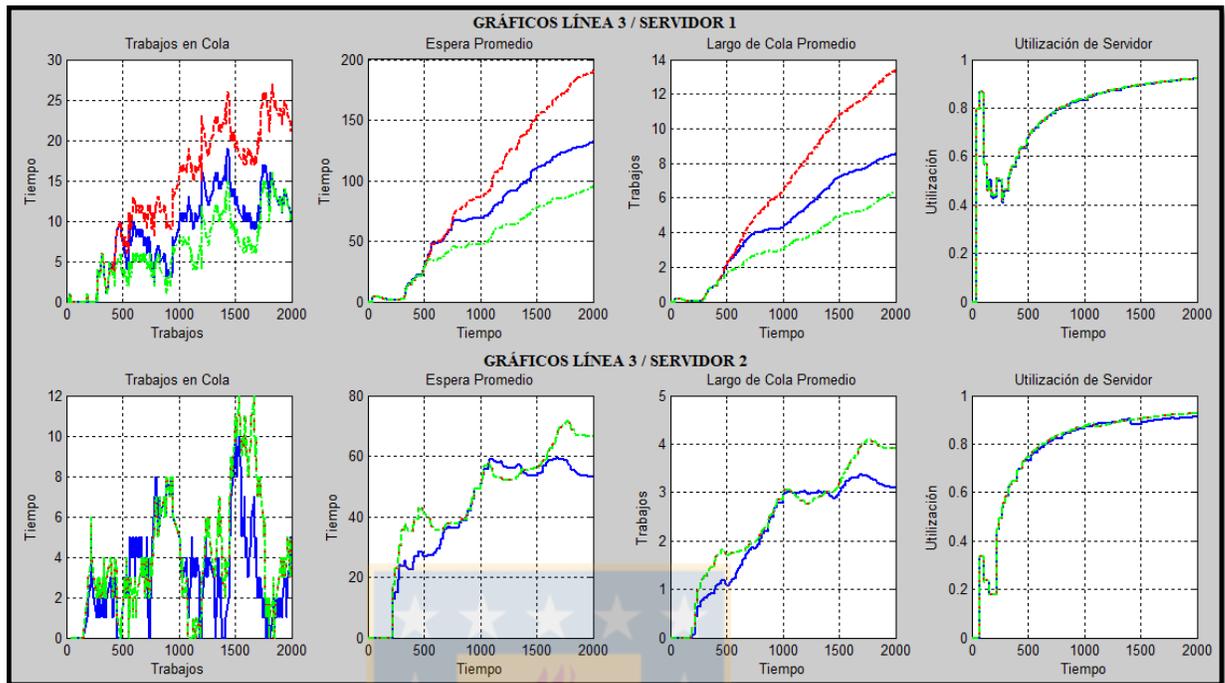
**Figura 79: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 7**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

En la Figura 80 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.



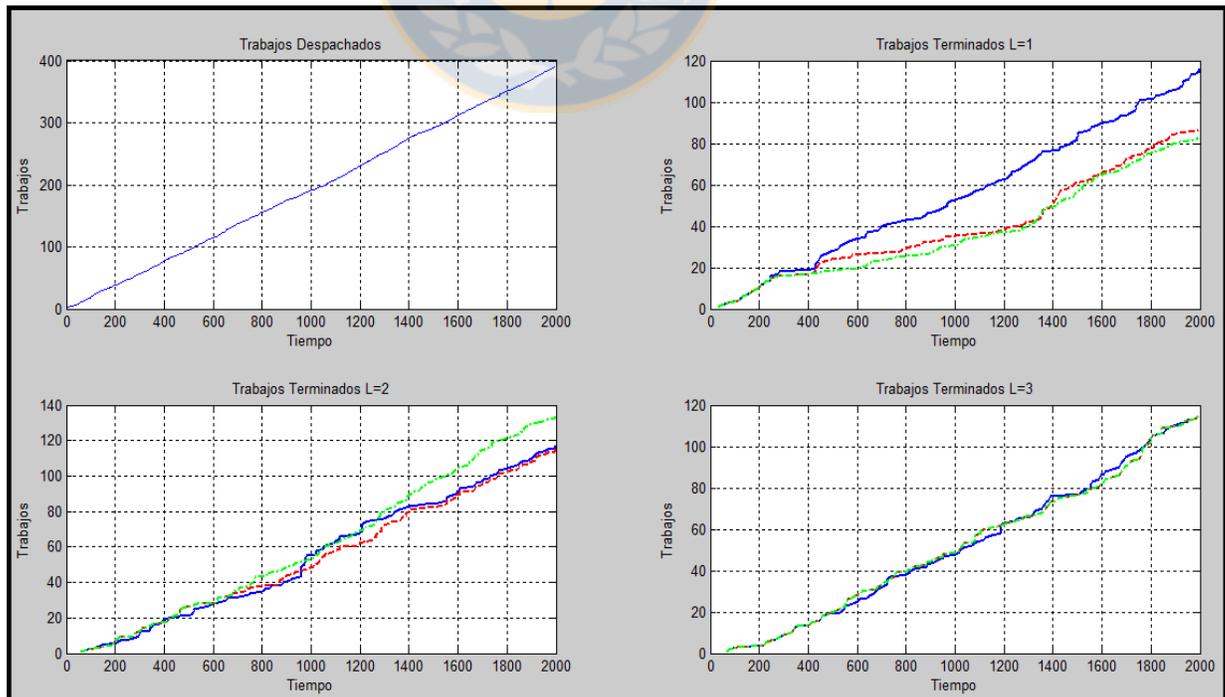
**Figura 80: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 7**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.) CRC-H con falla

En la Figura 81 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 81: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 7**  
 (—) Escenario Base (---) Ctrl Convencional con falla (— · —) CRC-H con falla

En la Figura 82 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 82: Trabajos despachados y terminados en L1, L2 y L3, Escenario 7**  
 (—) Escenario Base (---) Ctrl Convencional con falla (— · —) CRC-H con falla

### 3.3.4.17 A.3.9 Escenario 8: Falla en el servidor $S_{21}$ , con recuperación

En la Figura 83 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.

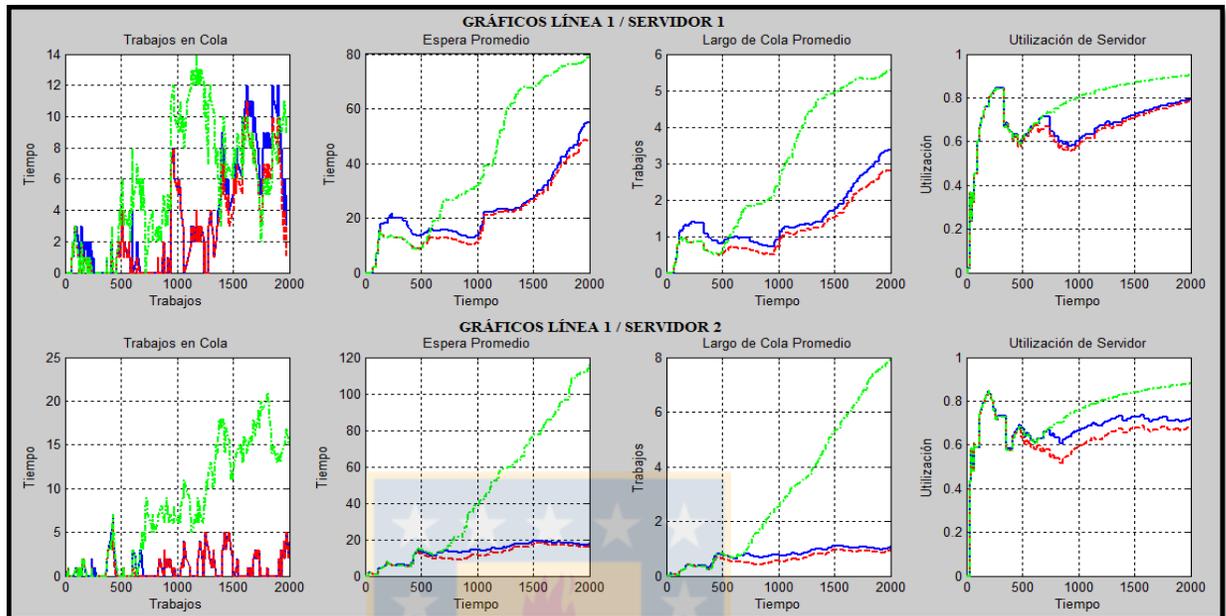


Figura 83: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 8  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.-) CRC-H con falla

En la Figura 84 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.

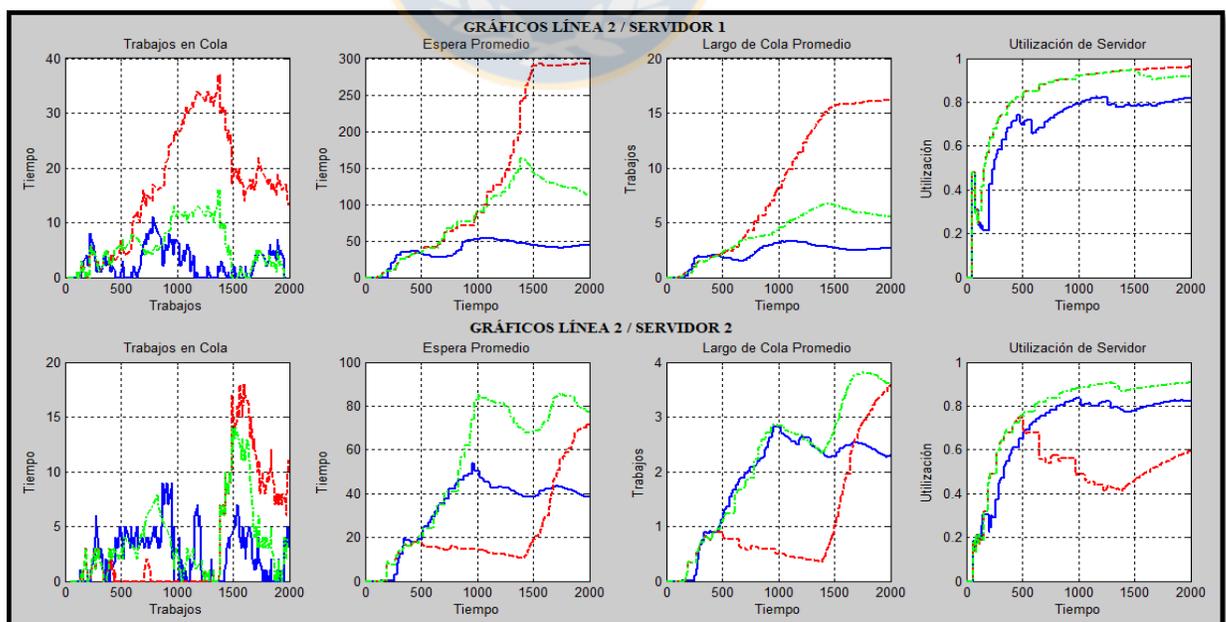
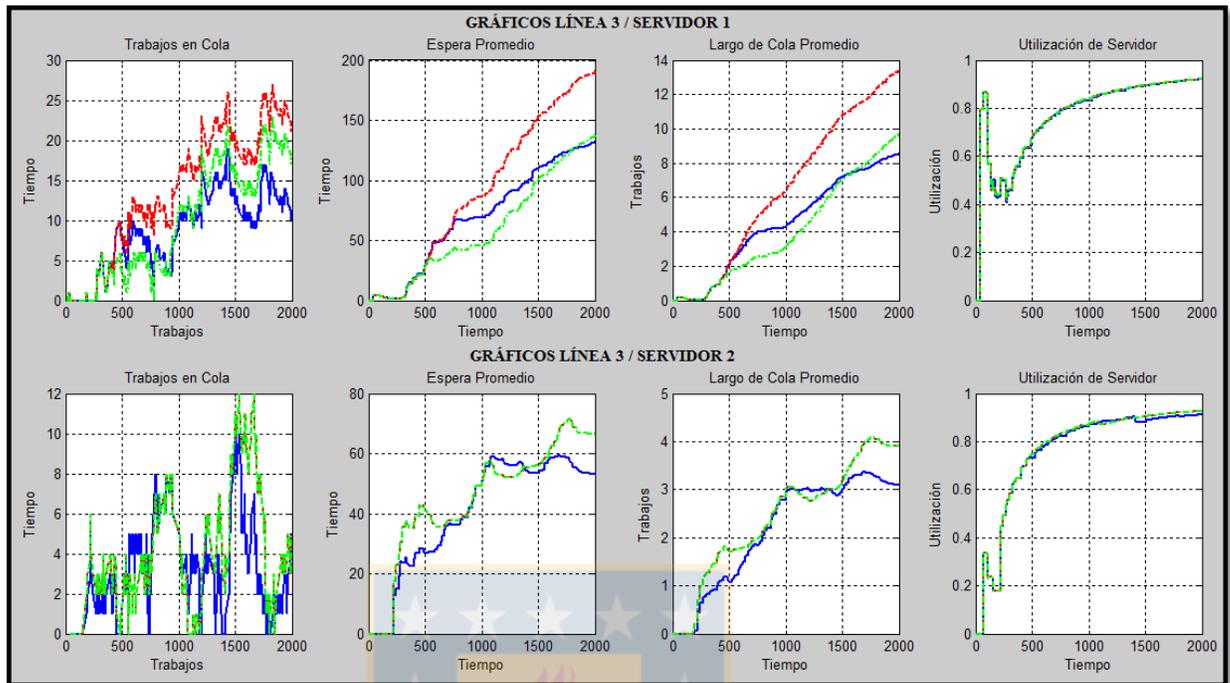


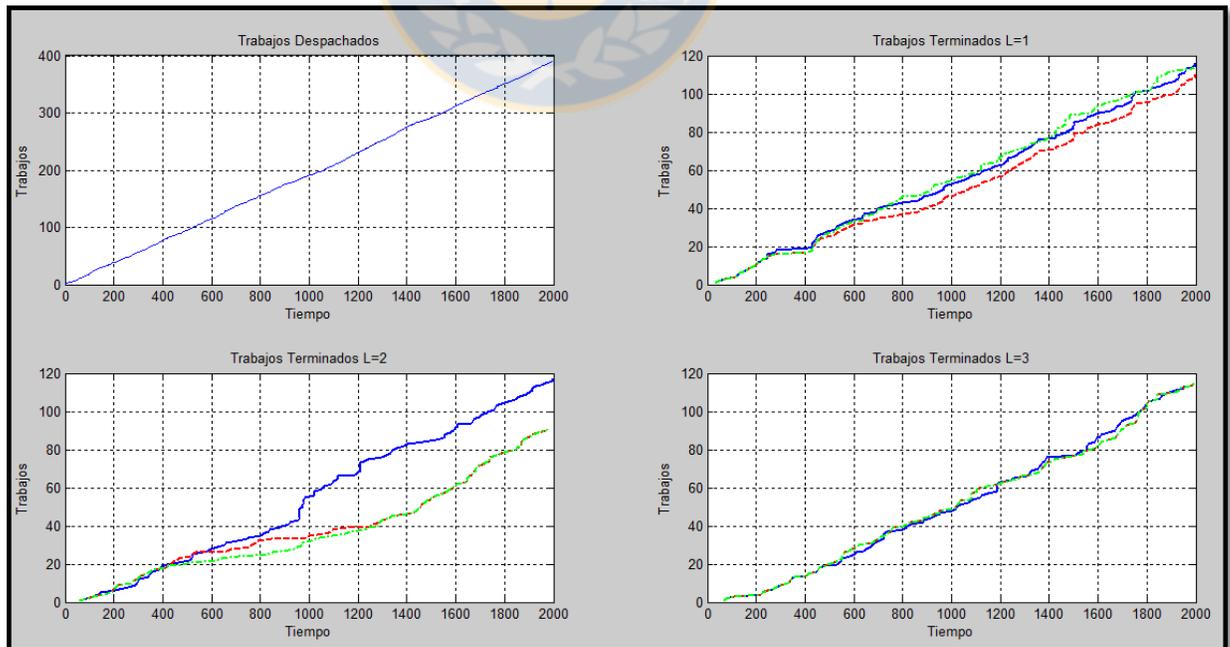
Figura 84: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 8  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.-) CRC-H con falla

En la Figura 85 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 85: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 8**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 86 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 86: Trabajos despachados y terminados en L1, L2 y L3, Escenario 8**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

### 3.3.4.18 A.3.10 Escenario 9: Falla en el servidor $S_{31}$ , con recuperación

En la Figura 87 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{11}$  y  $S_{12}$ , respectivamente.

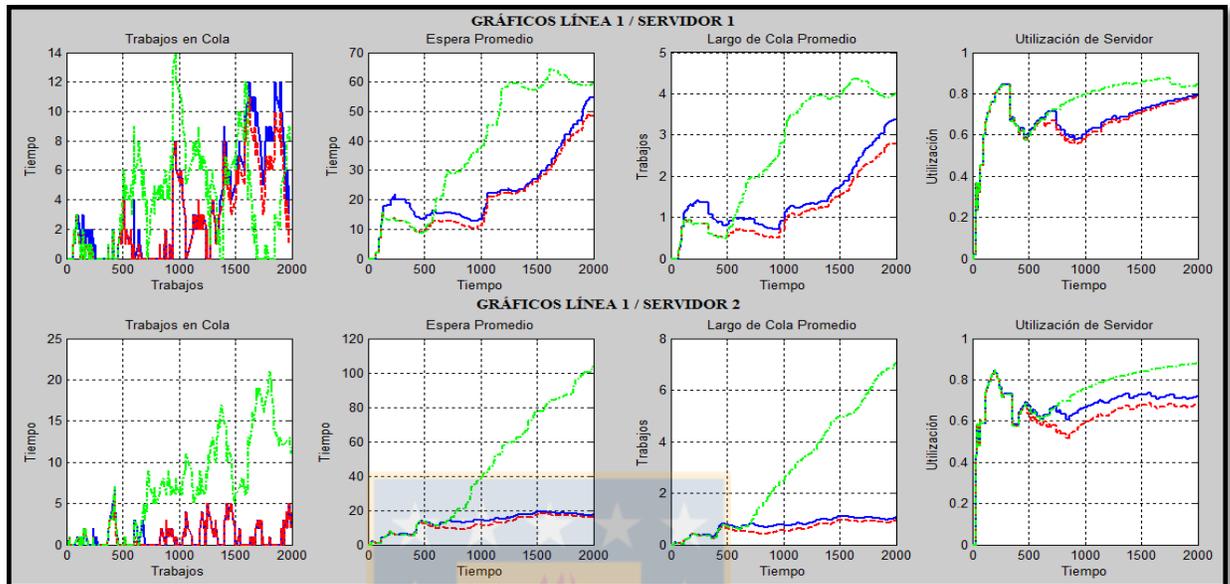


Figura 87: Gráficos Línea 1 / Servidor 1 y 2, Falla Escenario 9  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.-) CRC-H con falla

En la Figura 88 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{21}$  y  $S_{22}$ , respectivamente.

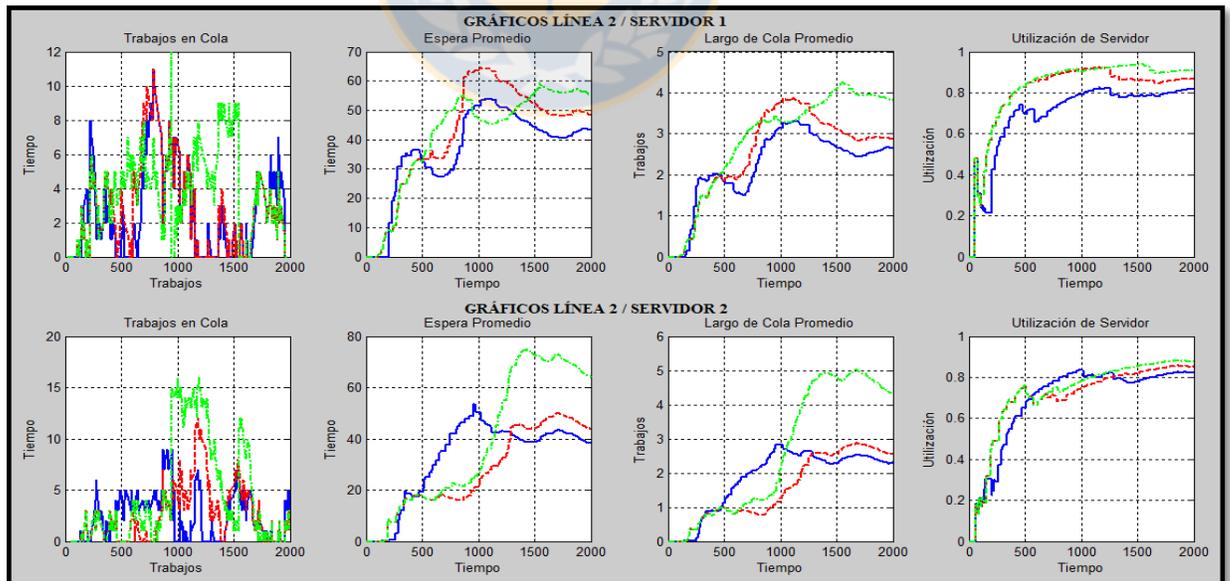
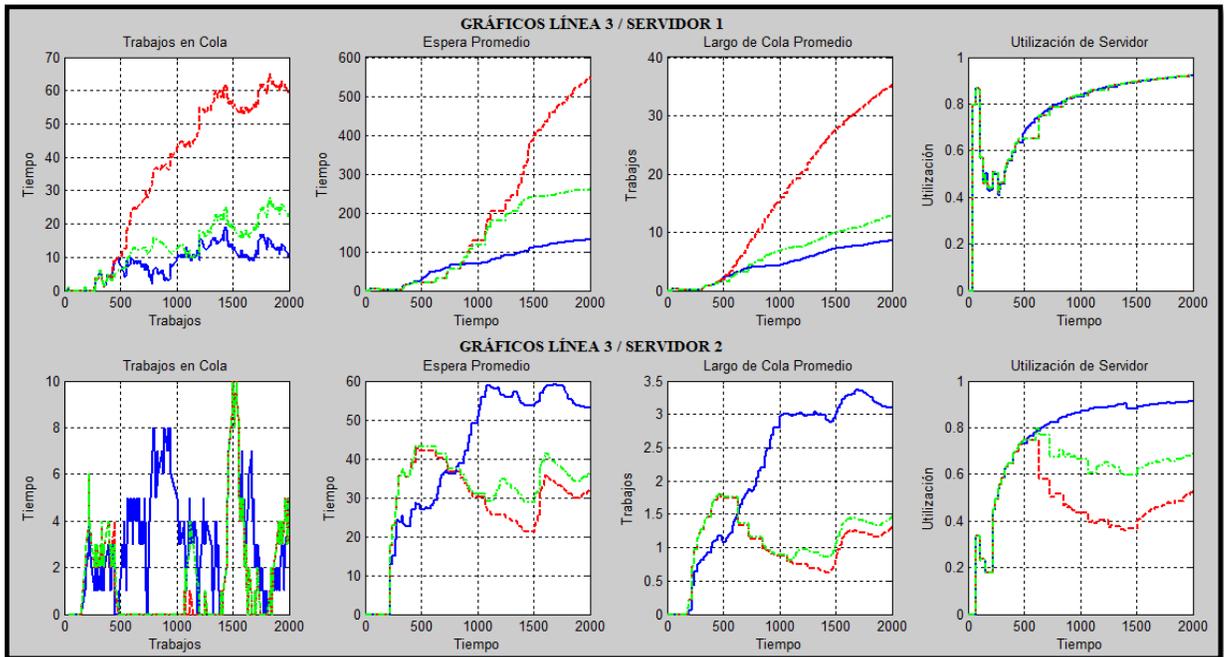


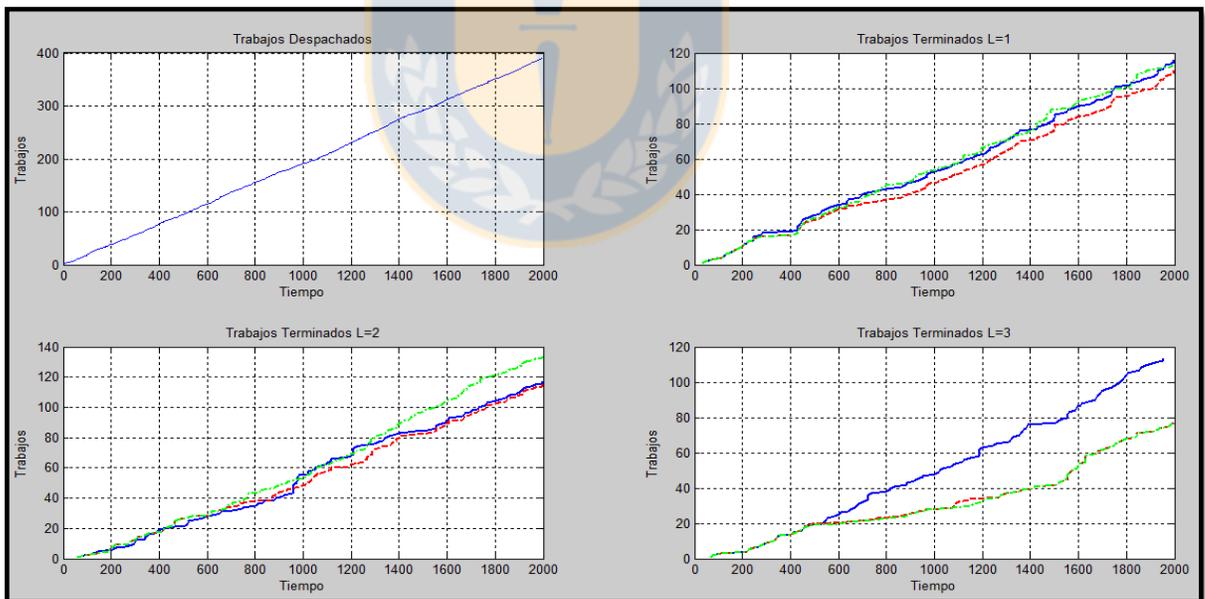
Figura 88: Gráficos Línea 2 / Servidor 1 y 2, Falla Escenario 9  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-.-) CRC-H con falla

En la Figura 89 se muestran las gráficas asociadas a trabajos en cola, espera promedio, largo de cola promedio y utilización asociados a los servidores  $S_{31}$  y  $S_{32}$ , respectivamente.



**Figura 89: Gráficos Línea 3 / Servidor 1 y 2, Falla Escenario 9**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla

En la Figura 90 se muestran las gráficas asociadas trabajos despachados (desde la fuente de emisión de trabajos), trabajos terminados en  $L1$ ,  $L2$  y  $L3$ .



**Figura 90: Trabajos despachados y terminados en  $L1$ ,  $L2$  y  $L3$ , Escenario 9**  
 (—) Escenario Base (---) Ctrl Convencional con falla (-.-) CRC-H con falla