



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas, Naturales y Agrimensura

Informe de Adscripción

Autonomic Computing



María Laura Regnet - L.U.: 27741

Prof. Director: Mgter. David Luis la Red Martínez

Asignatura: Sistemas Operativos

Licenciatura en Sistemas
Corrientes - Argentina

2005

A mi familia y profesores

Índice General

1	Nociones de Computación Autónoma	1
1.1	Introducción	1
1.2	Situación Problemática	3
1.3	Propuesta de Solución	4
1.3.1	Novedades de IBM para AC	6
1.3.2	El Nuevo Paradigma AC	7
1.3.3	Estándares Relacionados con AC	7
1.4	Beneficios Esperados	8
1.5	Características de los Sistemas de AC	9
1.6	Algunas Aplicaciones con Tecnología CA	11
2	Arquitectura de AC	15
2.1	Niveles de Maduración Autónoma	15
2.2	Arquitectura de Referencia de AC	16
2.2.1	Ciclo de Control	16
2.2.2	Administrador Autónomo	18
2.2.3	Recurso Administrado	19
2.2.4	Interfase de Manejabilidad	19
2.3	La Arquitectura AC y los Niveles de Maduración	20
2.4	Arquitectura AC Para Sistemas Personales (PC)	20
3	Introduc. y Tecnologías del AC Toolkit	27
3.1	Conceptos Generales	27
3.1.1	Tecnologías	28
3.1.2	Herramientas	28
3.1.3	Escenarios	28
3.1.4	Información y Documentación	29
3.2	Uso del AC Toolkit	29
3.2.1	Comprensión de Conceptos del AC	29

3.2.2	Tecnologías y Soluciones de AC Toolkit	29
3.2.3	Uso de Tecnologías del AC Toolkit	30
3.3	Tecnologías y Herramientas del AC Toolkit	31
3.4	AME	33
3.4.1	Introducción	33
3.4.2	Modelo de Recursos AME	33
3.5	Log and Trace Analyzer	34
3.6	Inst. de la Solución y Technolog. de Despliegue	35
3.7	Adaptador Genérico de Log	35
3.8	Consola de Soluciones Integradas	36
4	Escenarios e Instalación de AC Toolkit	39
4.1	Objetivos	39
4.2	Escenario de Determinación de Problema	39
4.3	Escenario de Instalación Automática	41
4.4	El Paquete AC Toolkit	42
4.4.1	AC Information	42
4.4.2	AME	43
4.4.3	RMB	43
4.4.4	Integrated Solutions Console	44
4.4.5	Solution Installation and Deployment	44
4.4.6	Generic Log Adapter and Log Trace Analyzer	45
4.4.7	Problem Determination Scenario	45
4.4.8	Solution Installation and Deployment Scenario Usando ISMP	47
4.4.9	Solution Installation and Deployment Scenario Using InstallAnywhere	47
4.5	Descarga del Paquete AC Toolkit	48
4.6	Instalación del Paquete AC Toolkit	48
4.6.1	Vista General de la Instalación	48
4.6.2	Prerequisitos	49
4.6.3	Plataformas Soportadas	50
4.6.4	Instalación del Paquete AC Toolkit	51
4.6.5	Desinstalación del Paquete AC Toolkit	52
4.7	Instalación de la ISC	52
4.7.1	Antes de Comenzar	52
4.7.2	Requerimientos Para la Instalación de la ISC	53
4.7.3	Instalación de la ISC	54
4.7.4	Desinstalar la ISC	63
4.8	Arrancar y Detener la ISC	63

<i>ÍNDICE GENERAL</i>	vii
-----------------------	-----

4.8.1 Secuencia Gráfica de Arranque y Detención de la ISC	64
4.9 Instalar el Plug-in de la ISC	67
4.10 Verificación de la Instalación	67
4.11 Resolver Problemas de Instalación	68

Bibliografía	71
---------------------	-----------

Índice de Materias	73
---------------------------	-----------

Índice de Figuras

1.1	Tecnologías para los emprendimientos, con innovación y productividad.	3
1.2	Dispositivos fuera de servicio.	4
2.1	Arquitectura de referencia de AC.	17
2.2	Arquitectura de un Elemento Autónomo.	21
2.3	Ejemplo de dos Jerarquías de Control Autónomo.	23
4.1	Ventana de bienvenida a la ISC.	55
4.2	Ventana de licencia de la ISC.	55
4.3	Ventana de información de la ubicación de la instalación de la ISC.	56
4.4	Ventana de cuentas de administrador de la ISC.	57
4.5	Nombre del host y localización del puerto.	57
4.6	Localizaciones de puertos nuevos - ventana 1.	58
4.7	Localizaciones de puertos nuevos - ventana 2.	58
4.8	Ventana de plug-ins del WSphere Application Developer.	59
4.9	Ventana de localización de instalación del WebSphere Studio Application Developer.	60
4.10	Ventana de localización e información de la instalación de la ISC.	61
4.11	Proceso de extracción de la ISC.	61
4.12	Proceso de finalización de la ISC.	62
4.13	Ventana del proceso final de la instalación de la ISC.	62
4.14	Windows.	64
4.15	Linux.	65
4.16	AIX.	65

Índice de Tablas

1.1	Comparación entre computación clásica y computación autónoma.	13
4.1	Espacio requerido por instalaciones Linux y AIX.	50
4.2	Espacio requerido por instalaciones Windows.	51

Capítulo 1

Nociones de Computación Autonómica

1.1 Introducción

Según Alfred North Whitehead “*la Civilización avanza al extender el número de operaciones importantes que podemos ejecutar sin pensar en ellas*”.

Esta cita hecha por el eminente matemático Alfred North Whitehead contiene tanto la cerradura como la llave a la próxima era de la informática. Implica el momento de superar el umbral sólo después de que los humanos ha podido automatizar cada vez más tareas complejas para alcanzar nuevos progresos.

Se cree que estamos actualmente sobre tal umbral en informática. Los millones de negocios, miles de millones de humanos que los componen, y billones de aparatos de los que se dependerá en todo momento, requieren los servicios de las industrias de IT para asegurar su funcionamiento. Y no es sólo cuestión de números. Es la complejidad de estos sistemas y la manera en que trabajan juntos la que crea una escasez de expertos en las IT para manejar todos los sistemas. Es un problema que crecerá exponencialmente, lo mismo que nuestra dependencia hacia dichas tecnologías [2] [7].

Para abreviar, no podemos esperar por mucho tiempo a los expertos en las IT.

Pero como Whitehead tan elocuentemente lo dijo hace casi un siglo, la solución estaría en la automatización, en crear una capacidad nueva donde importantes (y complejos) procesos informáticos puedan ejecutarse sin la necesidad de la intervención humana. El 15/10/2001, Paul Horn, vicepresidente senior de IBM Research, dirigió la Conferencia de Agenda, una reunión anual de las mentes tecnológicas más prominentes, que se desarrolló en Arizona, EE.UU. En su presentación y en un documento que distribuyó allí, sugirió una solución: *sistemas de computación que se autorregulen, de la misma manera en que nuestro sistema nervioso autónomo regula y protege nuestros cuerpos.*

Este nuevo modelo de computación se llamaba **autonomic computing** o **computación autónoma**. Las buenas noticias son que algunos componentes de esta tecnología ya están disponibles. Sin embargo, no existen todavía sistemas autónomos completos.

Ésta no es una solución propietaria de IBM. Es un cambio radical de la forma de manejar los negocios, la educación, el gobierno, etc., desarrollando, operando y manteniendo sistemas de computadoras. La *Computación Autónoma* (CA) exige un área nueva de estudio y una manera nueva de conducir los negocios.

Este *nuevo paradigma* cambia la definición fundamental de la tecnología computacional de un *paradigma centrado en los equipos*, a uno *centrado en los datos*. El acceso a datos de fuentes múltiples, distribuidas, además de las fuentes centrales tradicionales de almacenamiento, brindará a los usuarios el acceso a la información transparentemente, cuando y donde lo requieren. Al mismo tiempo, esta nueva visión de la informática hará necesario cambiar el enfoque de la industria computacional centrado en la velocidad de los procesos y en el almacenamiento, a un *enfoque de desarrollo distribuido en redes* que sean ampliamente *auto-gestionadas, auto-diagnosticadas, y transparentes* al usuario.

En este contexto es en el cual la innovación y la productividad se deben hacer presentes para facilitar la incorporación de las nuevas tecnologías y paradigmas que las sustentan, a los emprendimientos informáticos de la sociedad de la información y del conocimiento. Ver Fig. 1.1 de la pág. 3.



Figura 1.1: Tecnologías para los emprendimientos, con innovación y productividad.

1.2 Situación Problemática

Durante las pasadas dos décadas el desarrollo de la informática impulsada por la proliferación de equipos de computación ha crecido a ritmo exponencial. Este crecimiento fenomenal junto con el advenimiento de la Internet ha llevado a una nueva era de accesibilidad a otras personas, otros sistemas, y lo más importante, a más información [3].

Este boom de crecimiento ha llevado también a niveles de complejidad sin precedentes.

Para satisfacer las necesidades de los usuarios, los sistemas de computación se vuelven más complejos y más costosos de instalar y mantener. Hoy, los sistemas administradores deben contar con cientos de subsistemas y miles de parámetros, para tener corriendo los sistemas, y hay que enfrentarse a costos administrativos elevadísimos además de los de la adquisición de software y hardware.

La explosión simultánea de información e integración de tecnología en la vida cotidiana han traído nuevas demandas acerca de cómo las personas manejan y mantienen sistemas de computación. El número de puestos de trabajo de IT vacantes sólo en los Estados Unidos es de cientos de miles. Aún en

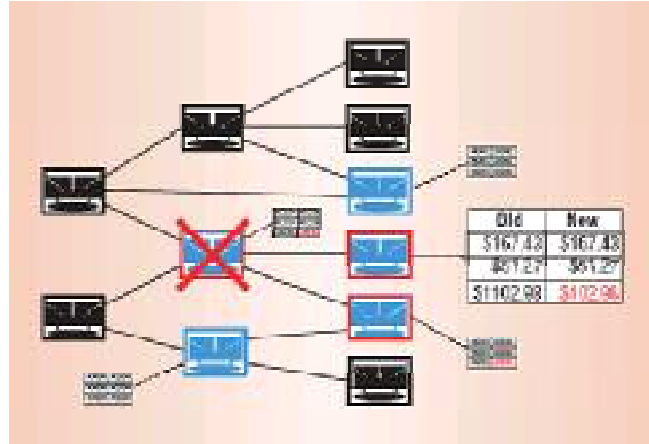


Figura 1.2: Dispositivos fuera de servicio.

tiempos de crecimiento económico incierto, la demanda de especialistas en IT se estima que crecerá por encima del 100% en los próximos seis años.

Como el acceso a la información se vuelve omnipresente debido a las PCs, los dispositivos manuales y los aparatos inalámbricos, la estabilidad de la actual infraestructura, los sistemas, y los datos, están en un cada vez más alto riesgo de sufrir salidas de servicio y daño general. Ver Fig.1.2 de la pag. 4.

IBM cree que estamos alcanzando rápidamente un momento umbral en la evolución de la informática en general y de la infraestructura asociada, el middleware, y los servicios que los mantienen. La creciente complejidad del sistema llega a un nivel más allá de la habilidad humana de manejarlo y asegurarlo.

Esta complejidad creciente con una escasez de profesionales experimentados de IT, apuntan hacia una inevitable necesidad de automatizar muchas de las funciones hoy asociadas con la computación.

1.3 Propuesta de Solución

La CA busca satisfacer necesidades con la mínima intervención de la administración humana. Los *sistemas autónomos* (SA) se adaptan a los cambios de

condiciones y a las cargas de trabajo, solucionan errores y fallas ellos mismos, y se preparan para defenderse de un próximo ataque.

Estos sistemas se manejan por sí mismos y reducen la complejidad desde el punto de vista de las tareas a cargo de los administradores y usuarios.

La solución propuesta por IBM mira el problema desde la perspectiva más importante: *el usuario final*. ¿Cómo quieren los clientes de IT que funcionen los sistemas de computación?. Quieren interactuar con los sistemas intuitivamente, y no quieren tener que estar directamente involucrados en su funcionamiento. Idealmente, a los usuarios de IT les gustaría sistemas informáticos bonitos, complejos y seguros, pero sin involucrarse en los aspectos de su manejo y mantenimiento.

IBM anunció nuevos servicios, software, adopción de estándares y programa de asociados para ayudar a las empresas a diseñar e implementar ambientes de computación autónoma de *autogestión* que pueden generar ahorros de tiempo de 30 a 50 por ciento en tareas de IT, según estimaciones de consultoría.

Las nuevas ofertas ayudarán a las empresas a transformarse según el modelo *on demand*, automatizando los procesos de computación y dotando a los sistemas informáticos de mayor capacidad de respuesta a las necesidades de los usuarios.

Permitirán administrar sistemas complejos construyendo y aplicando una arquitectura que los lleva hacia un ambiente autónomo, así como identificar y solucionar problemas en sus ambientes de IT multiproveedor (multiplataforma) y a crear sistemas que *automáticamente diagnostican la causa primaria de los problemas*, reduciendo el costo de las paradas de sistemas.

El nuevo software y los estándares que lo respaldan ayudarán a los desarrolladores y clientes a aplicar *tecnología de autogestión* a sistemas complejos.

La *inspiración* actual más directa para esta funcionalidad es el *funcionamiento autónomo del sistema nervioso central humano*. Los controles autónomos usan motores neuronales para enviar mensajes indirectos a los órganos en un nivel sub-consciente. Estos mensajes regulan temperatura, respiración y ritmo cardíaco sin utilizar el pensamiento consciente. Las implicaciones para la computación son inmediatamente evidentes; una red organizada de componentes computacionales “inteligentes” que dan lo que se requiere, cuando se lo requiere, sin esfuerzo físico ni mental consciente.

Desde que IBM introdujo el concepto de la CA en 2001, la compañía ha integrado numerosas características autonómicas en diversos productos, y cuenta con un importante grupo de productos, servicios y soluciones habilitadas para autogestión. Muchos están diseñados para automatizar la *localización de problemas de infraestructura* que, cuando es manual, puede consumir mucho tiempo. La firma de analistas Enterprise Management Associates estima que determinar la causa de un problema puede ocupar hasta el 50-80 por ciento del tiempo de un equipo de IT, mientras que la reparación en sí sólo ocupa el 15-20 por ciento restante.

1.3.1 Novedades de IBM para AC

Las principales novedades de IBM en el *mundo autonómico* son:

- *IBM Accelerator for Service Management for Problem Determination*, que permite a los clientes combinar, analizar y correlacionar información de eventos entre sistemas heterogéneos. Esto incluye adaptadores de log que pueden convertir datos de registro distintos a un formato común, proporcionando una única interfaz de usuario para tener una visión simplificada extremo a extremo y análisis y correlación de datos consolidados. Al poner su *Tecnología Autonómica Autoadministrable* al servicio de los clientes, se puede encontrar y reparar fallas de sistemas e interrupciones de servicio más rápidamente gracias a la automatización de los procesos.
- *Dynamic Infrastructure for my SAP Business Suite*, que proporciona una solución flexible que mejora la capacidad de los clientes de compartir recursos entre distintas aplicaciones SAP, acelerar la implementación de nuevas soluciones SAP, mejorar la utilización de sistemas y bajar el costo total de propiedad. Esta oferta aprovecha el *Tivoli Provisioning Manager*, que incluye tecnología autonómica autoadministrable.

También se dispone de un nuevo software autonómico en el *Toolkit de Computación Autonómica*, un recurso en línea donde los desarrolladores pueden implementar rápidamente funciones autoadministrables en sus aplicaciones y servicios. Respondiendo al feedback de los desarrolladores, el nuevo software autonómico los ayuda a llevar la tecnología autoadministrable a aplicaciones más grandes y complejas. Los desarrolladores pueden aprovechar código Java

en componentes para mayor flexibilidad y capacidades adicionales de filtros, a fin de acelerar el análisis y la determinación de problemas. Al incorporar estándares de la industria, la tecnología del Toolkit da soporte a aplicaciones heterogéneas.

1.3.2 El Nuevo Paradigma AC

Este nuevo paradigma computacional significa el diseño e implementación de sistemas de computación, software, almacenamiento y apoyo que deben exhibir los siguientes *principios básicos* desde la perspectiva del usuario:

- Flexibilidad:

El sistema podrá manipular datos a través de una plataforma y de unos dispositivos cuyo funcionamiento no sólo desconoce sino que le resulta indiferente.

- Accesibilidad:

La naturaleza del SA es tal que siempre está disponible.

- Transparencia:

El sistema ejecutará sus tareas y se ajustará a las necesidades del usuario sin que el usuario deba interiorizarse en las complejidades de su funcionamiento.

1.3.3 Estándares Relacionados con AC

IBM contribuye a impulsar los estándares en torno a la computación autónoma, colaborando con organizaciones de estándares y líderes de la industria. La especificación *Solution Installation* de IBM será considerada por un grupo de trabajo del organismo de estándares denominado *Solution Deployment Descriptor* (SDD), dentro de la *Organization for the Advancement of Structured Information Standards* (OASIS), que está desarrollando un método estandarizado para expresar las características de instalación de software requeridas para la gestión del ciclo de vida en ambientes distribuidos y multiplataforma.

Además, la especificación *Common Base Event* presentada por IBM ha sido un aporte crítico en el estándar *Web Services Distributed Management* (WSDM) recientemente ratificado por OASIS.

Los clientes y asociados ya están usando el formato *Common Base Event* y la tecnología autonómica *Solution Installation and Deployment*, demostrando el beneficio de la estandarización en torno a la CA. Por ejemplo, *Macrovision* incorporó la tecnología *Solution Installation and Deployment* en su producto *FLEXnet Publisher Installation Module*, recientemente anunciado. Usando esta oferta, la compañía de tecnología *SAS* tiene la capacidad de mantener, instalar, configurar y administrar software en forma automática y más eficiente, y espera una reducción significativa de sus costos de gestión de IT.

El software autonómico autoadministrable de IBM y los estándares que le dan soporte son elementos críticos de la estrategia de *Gestión de Servicios de IT de Tivoli*, que se focaliza en automatizar e integrar los procesos informáticos en toda la empresa u organización.

1.4 Beneficios Esperados

La CA se concibió para aminorar las demandas crecientes de personal altamente experimentado en las IT, reducir la complejidad y administrar la informática en una nueva era que aprovechará mejor su potencial para soportar niveles más altos de conocimiento en la toma de decisiones.

Los *beneficios inmediatos* incluirán una *dependencia reducida respecto de la intervención humana* para mantener sistemas complejos acompañada por una *disminución substancial en costos*. Los beneficios a largo plazo permitirán a los individuos, a las organizaciones y a las empresas, colaborar en la resolución de problemas complejos.

Los beneficios a *corto plazo* relacionados con las IT son:

- Menor experiencia y capacitación de los usuarios debido a sistemas más sensibles e inteligentes y de tiempo real.
- Disminución de costos al escalar (ampliar) su uso.
- Potencia, almacenamiento y costos escalables, optimizando el uso tanto para hardware como para software.

- Impulso al uso pleno de procesadores ociosos, incluso PCs hogareñas, mediante sistemas de computación en red (*grid computing*).
- Consultas en lenguaje natural permitirán respuestas más profundas y más exactas.
- Accesos indistintos a múltiples tipos de archivos. El uso de estándares abiertas permitirá que los usuarios manipulen datos de todo tipo de fuentes potenciales y re-asignarles el formato correcto “en vuelo”, es decir, al ser transmitidos de un dispositivo a otro.
- Estabilidad. Alta disponibilidad. Altos niveles de seguridad. Menos errores de sistema o de la red debido a la auto-reparación.

Los beneficios a *largo plazo*, que son los más significativos, incluyen:

- Realización de la visión de disponibilidad mediante el cambio de recursos disponibles a sistemas de alto rango.
- Incorporación (embebida) de capacidades autonómicas en clientes o dispositivos de acceso, servidores, sistemas del almacenamiento, middlewre, y la red misma.
- Construcción de sistemas autonómicos federados (multiplataforma).
- Administración de niveles de servicio extremo-a-extremo.
- Colaboración y resolución global de problemas. Los sistemas de computación distribuidos permiten compartir de una manera más inmediata la información y la potencia de proceso, impulsando el uso de complejos algoritmos matemáticos para resolver problemas.
- Procesos que requieren simulación masiva (pronósticos del tiempo, estudios médicos con proteínas, etc.) que precisan de procesadores que ejecuten 24/ 7 (24 horas los 7 días de la semana) por largos períodos de tiempo, como un año.

1.5 Características de los Sistemas de AC

Mientras la definición de CA probablemente se transformará mientras las tecnologías involucradas maduran, la lista siguiente sugiere que ocho características definen un *sistema de computación autonómico* (SAC):

1. Un SAC requiere “conocerse a sí mismo” (sus componentes también deben poseer una identidad de sistema). Dado que un “sistema” puede existir a muchos niveles, un SA requerirá un conocimiento detallado de sus componentes, estado presente, capacidad última, y de todas sus conexiones a otros sistemas, para gobernarse a sí mismo. Necesitará conocer la magnitud de sus “propios” recursos, esos que puede pedir prestado o presta, y esos que puede compartir o que debe gestionar sin compartir.
2. Un sistema de CA debe configurarse y reconfigurarse a sí mismo bajo condiciones variantes (y en el futuro, condiciones imprevisibles). La configuración del sistema o “setup” debe ocurrir *automáticamente*, así como ajustes dinámicos a esa configuración, para manipular mejor ambientes cambiantes.
3. Un sistema de CA nunca establece el statu quo (no permanece como está), siempre busca maneras de perfeccionar su funcionamiento. Supervisará sus partes componentes y el flujo de carga de trabajo para alcanzar metas predeterminadas del sistema.
4. Un sistema de CA debe ejecutar algo semejante a “curación” (reparación), debe poder recuperarse de rutinas y eventos extraordinarios que pueden causar en algunas de sus partes un funcionamiento defectuoso. Debe poder descubrir problemas o problemas potenciales, y entonces hallar una manera alternativa de usar los recursos o de reconfigurar el sistema, preservando su funcionamiento fácilmente.
5. Un mundo virtual no es menos peligroso que el mundo físico, así un sistema de CA debe ser un experto en auto-protección. Debe descubrir, identificar y protegerse a sí mismo contra varios tipos de ataques y mantener garantías globales de funcionamiento y de integridad.
6. Un sistema de CA debe conocer su entorno y el límite del contexto de su actividad, y actuar de acuerdo con ello. Encontrará y generará reglas acerca de cómo interactuar mejor con otros sistemas vecinos. Tomará recursos disponibles, igualmente negociará el uso por parte de otros sistemas de sus elementos subutilizados, cambiando a ambos, a sí mismo y a su ambiente en el proceso, en una palabra, adaptando.
7. Un sistema de CA no puede existir en un ambiente hermético. Dado que es independiente en su habilidad de manejarse a sí mismo, debe funcionar en un mundo heterogéneo e instrumentar estándares abiertos,

en otras palabras, un sistema de CA no puede, por definición, ser una solución propietaria, es decir dependiente de un determinado proveedor.

8. Un sistema de CA anticipará los recursos optimizados requeridos mientras mantiene oculta su complejidad. Debe ordenar los recursos de IT disminuyendo la brecha entre las metas de negocio o personales del usuario, y la implementación de recursos de IT necesarios para alcanzar esas metas, sin involucrar al usuario en esta implementación de recursos.

1.6 Algunas Aplicaciones con Tecnología CA

Algunas de las aplicaciones ya disponibles que han incorporado aspectos de la tecnología CA son las siguientes:

Autonomic Personal Computing (APC) es computación personal sobre plataformas de CA. El desafío de la APC es simplificar y acrecentar la experiencia del usuario, ayudándolo anticipándose a sus necesidades en un entorno complejo, dinámico e incierto.

IBM ha instalado sistemas de servidores *e-server* que incorporan tecnologías de CA que permiten la *autorrecuperación*, *autoconfiguración*, *autoprotección* y *autooptimización*. Esto brinda dos ventajas: reduce los gastos generales de gestión controlando costos de soporte, e incrementa la confiabilidad de un entorno heterogéneo de IT. El resultado brindará infraestructuras más flexibles que requieran menor gestión, mientras se minimizan los gastos administrativos. Teniendo en cuenta que administrar un servidor suele ser más costoso que el mismo servidor!!.

DB2 8.2: es la segunda versión DB2 que cuenta con funcionalidades de CA.

DB2 tiene la capacidad de tomar acciones tales como avisar cuando la base comienza a sufrir insuficiencias de espacio y agregar el espacio requerido por sí sola. Se debe a que el monitoreo permanente se hace en automático, el administrador de bases de datos puede dedicarse a tareas más creativas, tales como la arquitectura de una solución; en consecuencia, la organización requerirá de menos administradores de bases de datos, lo cual implica una reducción hasta 65% en los costos por concepto de mano de obra calificada.

WebSphere: software que forma la base sobre la que los programadores

construyen y administran sus aplicaciones. Equipado con más avances en la tecnología de Software basada en el lenguaje de programación Java y tecnologías de servicio de Internet, usadas para permitir que diferentes programas compartan información y trabajen juntos.

Concepto	Computación Clásica	Computación Autónoma
Auto-Configuración	Los centros de datos corporativos tienen múltiples fabricantes y plataformas. La instalación, configuración y la integración de los sistemas consumen tiempo y lleva a cometer posibles errores.	La configuración automatizada de componentes y sistemas siguen políticas de alto nivel. El resto de los sistemas se ajustan automáticamente y de modo sencillo.
Auto-Optimización	Los sistemas tienen cientos de seteos manuales y parámetros no lineales sintonizados cuya cantidad incrementa con cada versión.	Los componentes y sistemas continuamente buscan oportunidades de mejorar su propia eficiencia.
Auto-Reparación	La determinación de problemas, en sistemas complejos y grandes, puede requerir a un equipo de programadores durante semanas.	Los sistemas detectan, diagnostican y reparan automáticamente la localización del software y los problemas de hardware.
Auto-Protección	La detección y la recuperación de ataques y fallas en cascada es manual.	Los sistemas se defienden automáticamente de los ataques o fallas. Utilizan una advertencia temprana para anticipar y prevenir fallas del sistema.

Tabla 1.1: Comparación entre computación clásica y computación autónoma.

Capítulo 2

Arquitectura de AC

2.1 Niveles de Maduración Autonómica

Incorporar capacidades autonómicas en un entorno informático es un proceso evolutivo que hace posible la tecnología, pero es a fin de cuentas implementado por cada emprendimiento a través de la adopción de estas tecnologías, soporte de procesos, y habilidades [4]. Los productos, sistemas, y entornos IT pueden ser *clasificados* en los siguientes cinco *niveles de maduración* que muestran cómo una organización va evolucionando al hacer uso de las capacidades autonómicas, soportando procesos y funcionalidades:

1. Nivel 1: Basic: Básico: *Los profesionales de IT ejecutan todas las funciones manualmente:*
 - En el nivel básico, los profesionales en IT manejan cada elemento de la infraestructura independientemente y lo levantan, monitorean y eventualmente, lo restituyen.
2. Nivel 2: Managed: Administrado: *Los análisis y planes son hechos por los profesionales:*
 - En el nivel de administrado, las tecnologías del administración de sistemas pueden usarse para recolectar información desde diferentes sistemas sobre consolas más chicas, ayudando a reducir el tiempo que toma coleccionar y sintetizar información ya que el entorno de IT se vuelve más complejo.

3. Nivel 3: Predictive: Predictivo: *Los profesionales de IT eligen las recomendaciones y la implementación:*
 - En el nivel predictivo, *la capacidad de análisis se introduce en el sistema* para monitorear situaciones que se originan en el entorno, y analizar las situaciones para proveer los posibles cursos de acción. Los profesionales en IT hacen una determinación sobre qué curso de acción tomar.
4. Nivel 4: Adaptive: Adaptable: *Los profesionales de IT proveen políticas utilizadas para generar planes automáticamente:*
 - En el nivel adaptable, el entorno de IT puede tomar acciones automáticamente basado en la información disponible y en el conocimiento de “qué está pasando en el entorno”. Como los análisis y las tecnologías de los algoritmos van mejorando y el personal se vuelve cada vez más satisfecho con las capacidades de aviso y de predicción que brindan éstas tecnologías, los sistemas pueden evolucionar al nivel adaptativo.
5. Nivel 5: Autonomic: Autónomico: *Sistemas de política abierta:*
 - En el nivel autónomico, las políticas y objetivos de negocios gobiernan las operaciones de infraestructura de IT. Los profesionales en IT interactúan con las herramientas de tecnología autónomica para monitorear procesos de negocio, modificar objetivos, o ambos.

2.2 Arquitectura de Referencia de AC

Los conceptos autónomicos presentados en esta sección forman la base para un enfoque común y el conjunto fundamental de terminologías requeridas en la arquitectura de sistemas AC en un entorno heterogéneo.

2.2.1 Ciclo de Control

El arquitectura de referencia de AC comienza a partir de la premisa de que los atributos de la implementación de la auto administración originan un *ciclo de control (control loop) inteligente*. Este ciclo recopila información desde

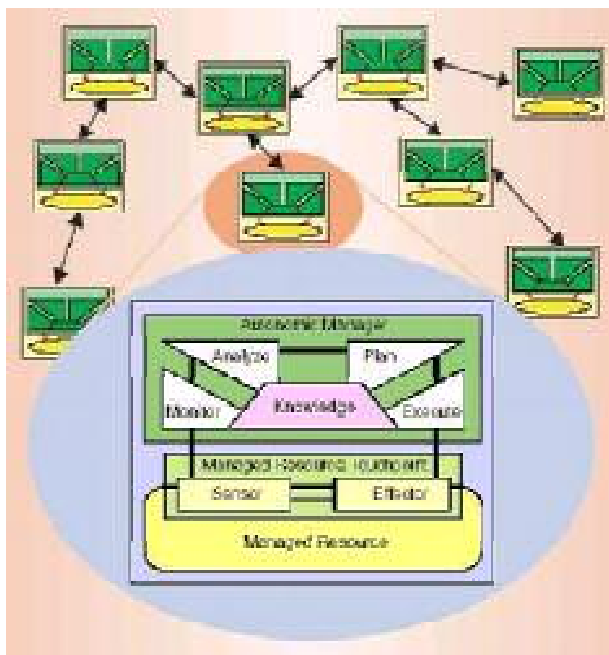


Figura 2.1: Arquitectura de referencia de AC.

el sistema, toma decisiones y entonces ajusta el sistema según sea necesario. *Un ciclo de control inteligente puede permitir al sistema disponer de atributos de auto-configuración, auto-reparación, auto-optimización y auto-protección descritos anteriormente.* La arquitectura describe dos tipos de componentes de los sistemas: un *administrador autónomo* y uno o más *recursos administrados*.

Un *administrador autónomo* es un componente que implementa un ciclo de control particular.

Un *recurso administrado* es lo que un administrador autónomo controla. Ver Fig. 2.1 de la pág. 17, donde se muestra la relación entre los diferentes componentes.

2.2.2 Administrador Autónomico

El *administrador autónomico* es un componente que implementa el “ciclo de control”. La arquitectura examina el loop en sus cuatro partes. Estas partes son:

- Monitor: Monitoreo: Alimenta el mecanismo que colecta, agrega, filtra, maneja y reporta detalles (métricas y topología) recolectados de un recurso administrado.
- Analize: Análisis: Provee los mecanismos de asociar y modelar situaciones complejas que permiten al administrador autónomico adquirir conocimiento acerca del entorno de IT y ayudar a predecir situaciones futuras.
- Plan: Plan: Brinda los mecanismos para estructurar la acción requerida para lograr propósitos y objetivos.
- Execute: Ejecución: Provee mecanismos que controlan la ejecución de un plan con consideraciones para su actualización sobre la marcha.

Las cuatro partes trabajan en conjunto para facilitar la funcionalidad del ciclo de control. Éstas utilizan y generan *conocimiento*. Este conocimiento se fundamenta en información conocida acerca del sistema y se incrementa a medida que el administrador autónomico aprende más sobre las características de los recursos administrados. El conocimiento es continuamente compartido entre las cuatro partes, llegando a decisiones tomadas por las partes, con más información. La Fig. 2.1 de la pág. 17 muestra una disposición estructural de las partes, no un ciclo de control.

La línea gruesa que conecta a las cuatro partes debería ser pensada como un bus de mensajes común más que como un ciclo de control. En otras palabras, pueden haber situaciones donde la parte “Plan” pueda requerir a la parte “Monitor” la recolección de más o menos información. Además podrían haber situaciones donde la parte “Monitor” pueda ordenar a la parte “Plan” a crear un nuevo Plan.

2.2.3 Recurso Administrado

El *recurso administrado* es un componente del *sistema administrado*. Puede haber un único recurso administrado (un servidor, servidor de base de datos, o un router) o una colección de recursos (una combinación de servidores, clusters, o aplicaciones de negocio).

Un administrador autónómico se comunica con un recurso administrado a través de la *interfase de manejabilidad*. Un *touchpoint* es la *implementación de la interfase de manejabilidad por un recurso administrado específico*. Por ejemplo, una base de datos puede implementar un touchpoint para comunicarse con un administrador autónómico.

2.2.4 Interfase de Manejabilidad

La *interfase de manejabilidad* entre un administrador autónómico y un recurso administrado se organiza dentro de las operaciones de “*sensor*” y “*effector*”.

En términos más simples, las operaciones *sensor* son típicamente usadas para *transmitir* eventos o propiedades a un administrador autónómico, mientras que las operaciones *effector* son típicamente usadas para *causar* algún tipo de *cambio* en un recurso administrado, tales como alterar datos o fijar valores de propiedades. Las operaciones *sensor* y *effector* se organizan dentro de un conjunto de estilos de interacción que formaliza y define cómo interactúan un administrador autónómico y su recurso administrado. Cada una de las operaciones *sensor* y *effector* pueden tener dos estilos de interacción:

- Sensor retrieve-state: Sensor recuperación-estado.
- Sensor receive-notification: Sensor recepción-notificación.
- Effector perform-operation: Effector realización-operación.
- Effector call-out-request: Effector llamada-cierre-requerimiento.

Los *estilos de interacción* se diferencian por cómo el administrador autónómico o el recurso administrado hacen el *primer contacto*. Tanto en el estilo de interacción *sensor retrieve-state* como en el *effector perform-operation*, el

administrador autónomico hace el primer contacto. En los estilos de interacción *sensor receive-notification* y *effector call-out-request*, es el recurso administrado el que hace el primer contacto. La combinación de las operaciones *sensor* y *effector* forma la interfase de manejabilidad que está disponible para un administrador autónomico.

Como muestra la Fig. 2.1 de la pág. 17, las líneas negras conectan la sección *sensor* y *effector* del diagrama; la arquitectura promueve la idea de que las operaciones *sensor* y *effector* están conjuntamente vinculadas. Por ejemplo, un cambio en la configuración que ocurra a través de un *effector* debería reflejarse como una notificación de cambio en la configuración a través de la interfase *sensor*.

2.3 La Arquitectura AC y los Niveles de Maduración

Como se describió en *Niveles de Maduración Autónoma*, los niveles muestran que los atributos de auto-administración se logran en una manera evolutiva de introducirse en todos los aspectos de un sistema. Como un ejemplo, diferentes partes de un administrador autónomico podrían implementarse a cada nivel de maduración.

Las partes *Monitor* y *Execute* del administrador autónomico podrían implementarse en los niveles *Basic* y *Managed*. Así, a estos dos niveles, los profesionales de IT serían responsables por la ejecución de las funciones de analizar y diseñar componentes. La parte de Análisis de un administrador autónomico puede ser suministrada en el nivel *Predictive* de maduración. En este nivel, el profesional en IT sería responsable de las funciones de *Plan*. En los niveles *Adaptive* y *Autonomic*, todas las partes del administrador autónomico se implementan de manera que el profesional en IT podría delegar el trabajo al sistema.

2.4 Arquitectura AC Para Sistemas Personales (PC)

Se detallará a continuación una arquitectura para *sistemas de computación autónomica personal* [7]. La arquitectura de un sistema autónomico incluye a la de un sistema de computación personal, comienza con la arquitectura

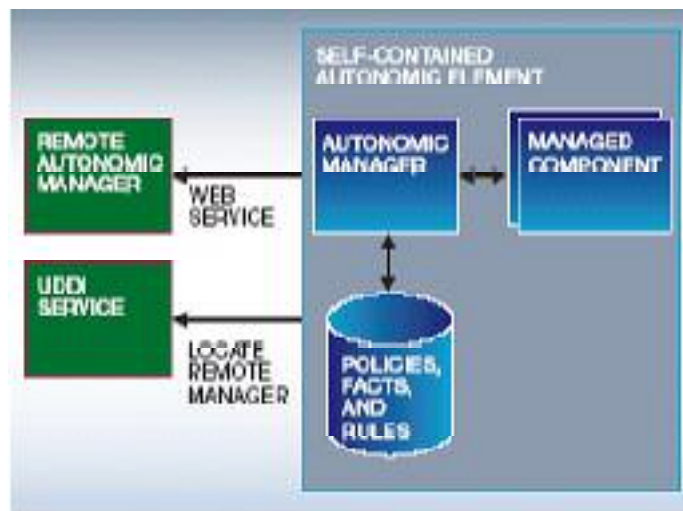


Figura 2.2: Arquitectura de un Elemento Autónomo.

general para sistemas autónomos. El bloque de construcción de un sistema autónomo está descrito en la Fig. 2.2 de la pág. 21.

La figura indicada muestra la arquitectura de un *elemento autónomo* (AE). Cada AE consiste de un *administrador autónomo* (AM) y un conjunto de componentes administrados. Cada componente administrado es responsable de comunicarse con eventos y otras medidas del AM local. A su vez, basado en la entrada recibida desde cada componente administrado, el AM toma decisiones teniendo en cuenta su política, los hechos y las reglas (almacenadas localmente en una base de datos) y comunica las directivas y sugerencias al componente administrado.

La figura anterior marca una distinción entre el comportamiento autónomo auto-controlado (dentro de la caja gris mayor de la derecha) y el administrador autónomo que comprende comunicaciones explícitas con un administrador remoto. Las interfases entre un AM y su componente administrado son una parte importante de la arquitectura. Para sistemas de computación personal basados en Windows, un amplio conjunto de éstas interfases están representadas por el *Windows Management Instrumentation* (WMI). La interfase entre un AM remoto y un AE no está actualmente estandarizada.

Esta interfase debe ser descubierta y dinámicamente contenida en el so-

porte de auto-configuración de sistemas autónomos, además debe ser segura y confidencial. La figura mencionada muestra una implementación de un administrador autónomo remoto de un *servicio Web* (Web service) mediante el servicio de registro *Universal Description, Discovery, and Integration* (UDDI).

Se aprecia a los servicios Web como una tecnología fundamental porque ellos proveen métodos estándares para localizar, comunicar (vía XML), construir, e interactuar con servicios basados en sistemas de redes. Pero debido a que los sistemas personales frecuentemente son móviles y ocasionalmente se desconectan, la interfase debe soportar también el modo de uso “desconectado” (off line).

La Fig. 2.3 de la pág. 23 muestra la arquitectura de un sistema autónomo que consiste de elementos autónomos conectados a otro en modo local, par, y niveles de sistemas de redes. Los recursos se muestran como cajas, los AMs con forma de diamante, los grupos de pares (entidades similares) como elipses, y los servidores de recursos físicos y clientes como círculos. Las flechas representan el control ejercido por los AMs (ej., S controla a W).

En el nivel local hay un solo AM (ej., A) que es capaz de tomar decisiones independientemente. A nivel de pares (grupo), cada AM interactúa y comparte conocimiento e información con sus pares y puede actuar de modo cooperativo, como si se estuviera en presencia de un administrador autónomo virtual (ej., W).

La idea de un *administrador autónomo virtual* es tener los participantes en un grupo de pares que alcance un cierto comportamiento autónomo como si fuera dirigido por una instancia local o remota de un administrador autónomo, aunque ninguno esté presente.

El comportamiento es una consecuencia del consenso local obtenido compartiendo conocimiento entre los pares, y actuando localmente basados en ese conocimiento. Los elementos de una arquitectura de pares, descubrimiento, pregunta, y unión, soportan este proceso.

Una de las *aplicaciones centrales de la Computación Autónoma Personal* es cómo los administradores autónomos en un cierto nivel más alto en el sistema, ejercen control sobre los elementos de nivel inferior, qué elementos autónomos son controlados, cuánto (y cuán rápidamente) se ejerce el control, y qué clase de control se realiza.

Se pueden identificar dos tipos de control: *delegación* y *dirección*.

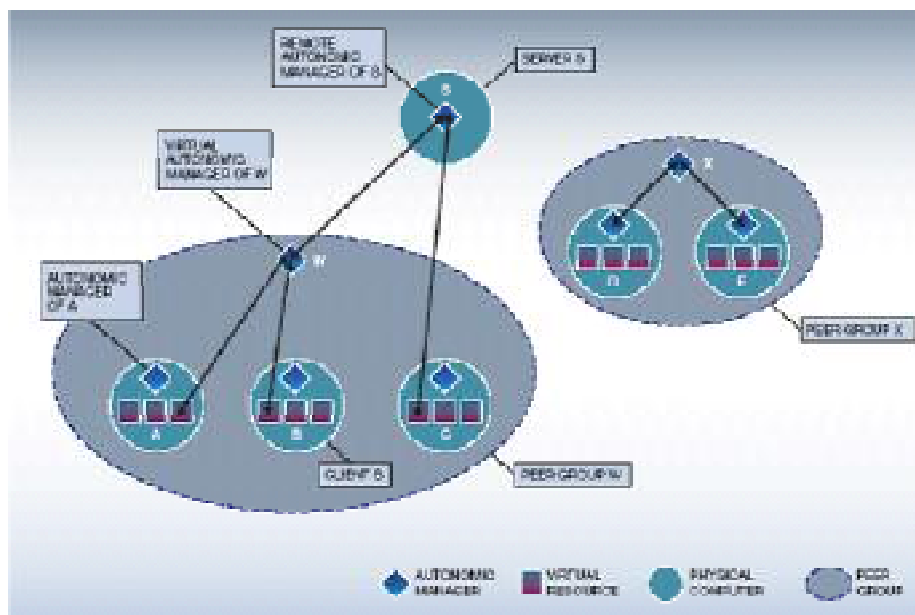


Figura 2.3: Ejemplo de dos Jerarquías de Control Autónomo.

En la *delegación*, un administrador autónomo local pasa el control de algunos de los recursos que maneja a un superior. Por defecto, el control de todos los recursos locales es delegado al AM local.

En la *dirección*, un administrador autónomo local recibe la información (ej., políticas) de su superior y las implementa con respecto a sus propios recursos. Solamente un AM está siempre en directo control de un recurso.

Las computadoras personales de hoy manejan todos sus recursos con un solo administrador de su sistema operativo.

Es ventajoso soportar un modelo en el cual el control sobre algunos recursos se pueda delegar a otro administrador, esto es posible con la virtualización del cliente.

Si se piensa en las cajas de la Fig. 2.3 de la pág. 23 como máquinas virtuales, se puede ver que dos de las máquinas virtuales del cliente A están bajo control local completo, mientras que la tercera está bajo control del grupo de pares. El cliente tiene dos máquinas virtuales controladas localmente y una administrada centralmente, quizás por el departamento de soporte y servicio

de IT de las organizaciones.

En el caso de desconexión ocasional de un administrador autonómico, una estrategia estática sencilla hace que el administrador remoto proporcione la *dirección* de la política pero no el control en tiempo real. Un ejemplo viene de la administración de la seguridad.

Un administrador de seguridad global remoto puede conocer los usos inter e intra-organización (ej., un nuevo estilo de ataque) que un cliente desconectado no puede descubrir hasta el momento crítico de la primera reconexión.

Por el momento, existe una carrera entre la actualización del administrador remoto y el ataque, sugiriendo que las precauciones especiales las tome el cliente para obtener con seguridad la directiva más reciente de seguridad antes de recomenzar su política de seguridad normal local.

Tales estrategias soportan el estilo tradicional de la computación personal de autonomía local considerable. Ahora se pueden discutir dos cuestiones que se consideran fundamentales para el éxito de esta arquitectura: *seguridad* y *privacidad* por una parte, y *estabilidad* por la otra.

Seguridad y privacidad: La seguridad y la privacidad son atributos de sistema críticos. Para que una máquina solicite información confidencial de otra máquina, necesita ser capaz de identificarse con seguridad.

Hay varias maneras de poder hacer esto, pero la más fácil (y probablemente la más segura) es usar una par de clave privada/pública. La TCPA (*Trusted Computing Platform Alliance*) provee una manera ideal de obtener claves privadas seguras para la identificación a nivel de máquina. Cada sistema puede entonces identificar cualquier otra máquina usando claves registradas mantenidas en una base de datos, como en el Administrador de Políticas de Tivoli IBM, o usando certificados.

Una vez que finalizada la identificación, es necesario establecer una conexión segura entre las computadoras. La manera estándar de hacer esta conexión, es mediante el *Internet Protocol Security* (IPSec) (Protocolo Seguro de Internet) pero también sirve el modelo cliente/servidor autenticado *Secure Socket Layer* (SSL). Este provee peticiones autenticadas, transmisión confidencial de los datos, e integridad de los datos de respuesta. Los ejecutables deben ser marcados para que puedan verificarse como provenientes de un recurso confiable y corran en un entorno seguro.

Un más alto nivel de administrador autónomico no debería enviar información a un administrador subordinado sin serle requerido. Cuando un sistema está en modo de transmisión, el receptor necesita ser capaz de acelerar el funcionamiento del sistema emisor para evitar el ataque de denegación de servicio.

Para que una máquina publique la información, necesita poder determinar si la información es confidencial. La información no confidencial debe ser identificada explícitamente, de otra manera, todo se asume como confidencial.

Estabilidad: La meta de un sistema personal autónomico es exhibir un comportamiento autónomico y a la vez útil para su usuario final. Cuando los componentes autónomicos se ponen juntos en un sistema, no es un hecho que el sistema en su totalidad exhiba comportamiento estable.

Por ejemplo, una computadora personal puede hacer una evaluación local de que un vínculo de comunicaciones compartido tiene gran ancho de banda, basada en las características de los medios, y puede iniciar una actividad (transmisión) de consumo de ancho de banda. Si otras computadoras personales toman la misma decisión, el enlace puede saturarse y llegar rápidamente a ser inutilizable para todos.

Una estrategia más conservadora, por ejemplo, es diferir actividades de consumo de ancho de banda hasta tener un registro (información) del funcionamiento del enlace, y entonces procurar un uso más agresivo del recurso; esto puede lograr un comportamiento total mejor.

En general, se puede imaginar que el comportamiento autónomico constructivo será acotado por el contexto y la política. Qué contexto y qué políticas (y cómo representarlas y mantenerlas) son puntos importantes para futuras investigaciones.

Capítulo 3

Introducción y Tecnologías del Autonomic Computing Toolkit

3.1 Conceptos Generales

El *Autonomic Computing Toolkit* es una colección de tecnologías, herramientas, ejemplos, escenarios y documentación que se diseñó para usuarios que quieran aprender, adaptarse, y desarrollar capacidades autonómicas en sus productos y sistemas.

El primer lanzamiento de *AC Toolkit* proporciona la primera fase de tecnologías AC que habilita el desarrollo de capacidades autonómicas. El contenido de *AC Toolkit* puede dividirse en cuatro categorías principales:

- Tecnologías.
- Herramientas.
- Escenarios.
- Información y documentación.

3.1.1 **Tecnologías**

Las tecnologías proporcionadas en *AC Toolkit* pueden ser usadas para desarrollar o mejorar ciertas capacidades en productos y sistemas. Estas capacidades incluyen determinación de problemas, sistemas de administración comunes, y soluciones de instalación y despliegue [4].

La determinación de problemas y capacidades autonómicas, pueden desarrollarse con *Autonomic Management Engine (AME)*, el *Generic Log Adapter*, y el *Log and Trace Analyzer Tool*. La *Integrated Solutions Console* (Consola de Soluciones Integradas) es utilizada para construir, de forma efectiva, las capacidades de administración de sistemas más comunes.

El *dependency checker* es una tecnología que provee capacidades autonómicas para soluciones de instalación y despliegue.

3.1.2 **Herramientas**

Además de proporcionar estas tecnologías, el *AC Toolkit* provee las herramientas necesarias para personalizar las mismas con el objetivo de que las soluciones se puedan crear adaptándolas a las necesidades específicas de cada usuario. Herramientas tales como *Integrated Solutions Console*, *Resource Model Builder (RMB)*, y el *Adapter Rule Editor Tool* están incluidos para asistir en la creación de estas soluciones personalizadas.

El *AC Toolkit* también provee programas de análisis de registro (log) para varios productos IBM. Estos programas de análisis, junto con los que el usuario podrá desarrollar para sus propios productos con el *Adapter Rule Editor Tool*, pueden ser usados para detectar problemas complejos en un entorno de sistema.

3.1.3 **Escenarios**

También están incluidos los escenarios, que muestran cómo las tecnologías trabajan de forma conjunta y cómo pueden ser usadas en soluciones reales. Todos los escenarios de *AC Toolkit* están realizados usando tecnologías y herramientas disponibles en *AC Toolkit*.

El primer lanzamiento de *AC Toolkit*, incluye un escenario para la de-

terminación de problemas, que realiza *auto-reparación* y dos escenarios de instalación automatizados que realizan tareas de *auto-configuración*.

3.1.4 Información y Documentación

El *AC Toolkit* también se focaliza en educar usuarios en computación autónoma.

Se proporcionan detalles de la tecnología individual y documentación acerca del uso de las herramientas, así como documentación para ayudar a comenzar el desarrollo autónomo de soluciones personalizadas para los productos del usuario.

3.2 Uso del AC Toolkit

Esta sección detalla lo que se puede realizar usando AC Toolkit, y cómo hacerlo.

3.2.1 Comprensión de Conceptos del AC

Antes de intentar desarrollar una solución autónoma, lo mejor es tener un buen conocimiento de los conceptos detrás de la computación autónoma, para lo cual es necesario releer los capítulos previos.

3.2.2 Trabajo Conjunto de las Tecnologías de AC Toolkit Para Lograr Soluciones de Auto-Gestión

Después de comprender los conceptos básicos de computación autónoma, se pueden utilizar los escenarios en *AC Toolkit*, para ver cómo estas tecnologías trabajan conjuntamente para lograr una solución auto-gestionada.

El *AC Toolkit* incluye también tres escenarios: un escenario para la *determinación de problemas*, que realiza auto-reparación y dos escenarios de *instalación* automatizados que realizan tareas de auto-configuración.

El escenario para la determinación de problemas representa un sistema

de auto-reparación simple que usa un *lazo* (loop) de control inteligente para recolectar información del sistema, analizarla, plantear respuestas apropiadas, y hacer los ajustes necesarios para resolver problemas.

Se incluye también documentación adicional en el paquete de escenarios. Los escenarios de instalación autónoma demuestran cómo soluciones de instalación autónomas y tecnologías de despliegue, se pueden usar para mejorar la comprensión e instalación de una aplicación web sencilla. Se proveen dos escenarios, con capacidades similares, pero cada uno usa diferentes paquetes de instalación de software del proveedor.

Esto demuestra una *versatilidad* y *flexibilidad* en la instalación de soluciones autónomas y el despliegue de tecnologías incluidas en *AC Toolkit*.

3.2.3 Uso de Tecnologías del AC Toolkit Para Desarrollar Soluciones de Usuario

Tan pronto como se haya adquirido un conocimiento sobre conceptos autónomos, y se haya experimentado cómo las tecnologías de *AC Toolkit* trabajan en forma conjunta, para lograr una solución auto-controlada, se puede comenzar a desarrollar capacidades de auto-control o auto-manejo para los productos propios, y ayudar a los clientes a incrementar el nivel de maduración autónoma en sus entornos IT.

El incremento de la maduración autónoma de los clientes significa que los sistemas tendrán menos *tiempo muerto* y requerirán de menos administración humana e intervención.

Esto puede proporcionar reducciones significativas en los costos de administración que pueden ser aplicados al desarrollo de soluciones que aumenten el valor de los negocios. Las tecnologías disponibles en *AC Toolkit* pueden ser usadas para posicionar los productos y soluciones para ayudar a los clientes a adquirir un mayor nivel de maduración en sus entornos IT:

- La *Integrated Solutions Console* (Consola de Soluciones Integradas), provee la infraestructura necesaria para ayudar a consolidar la interfase de usuario en un entorno IT heterogéneo. Esto puede ser usado para desarrollar *sistemas gestionados* más eficientes (nivel 2). Al agregar otras capacidades de gestión del sistema que corran dentro de la *Integrated Solutions Console*, se pueden lograr niveles de maduración mayores.

- El *Log and Trace Analyzer (LTA)* puede ser usado para lograr maduración nivel 2 (*managed*), y los atributos de auto-reparación del nivel 3 (*predictive*). El *LTA* provee una visión correlacionada de los eventos que ocurren en el sistema. Esto permite al personal de IT un mejor manejo de su entorno (nivel 2). Se pueden presentar al personal de IT la configuración y la adecuación al usuario de la base de datos de análisis de síntomas (problemas) y la capacidad de correlación disponible con el *LTA*, las directivas de resolución de problemas y opciones basadas en la ocurrencia de eventos. Esto logra un nivel de auto-manejo (nivel 3).
- El *dependency checker* (verificador de pre-requisitos) disponible en el *AC Toolkit*, puede ser usado para mejorar el proceso de instalación de los productos de software. El nivel 3 de función (*predictive*) se puede alcanzar mediante el desarrollo de procesos de instalación que incorporen esta tecnología. El *dependency checker* señalará dependencias (pre-requisitos) que falten, y se podrá programar el proceso de instalación para informar al usuario de esta situación, de esta manera, se podrá tomar alguna acción para correctiva.
- Los componentes AME de *AC Toolkit* pueden ser usados para desarrollar un nivel de adaptación del auto-manejo (nivel 4). Se puede utilizar el RMB disponible para desarrollar capacidades efectivas de auto-manejo para usarlas con AME.

3.3 Tecnologías y Herramientas del AC Toolkit

Las tecnologías y herramientas contenidas en el *AC Toolkit* pretenden asistir a los desarrolladores de productos para comenzar con el desarrollo de capacidades autónomas en sus productos, para ayudar a sus entornos de IT a lograr más altos niveles de maduración.

Las *tecnologías* en el *AC Toolkit* pueden dividirse en cuatro categorías, como muestra a continuación:

- Autonomic Managers:
 - Log and Trace Analyzer Autonomic.
 - Management Engine.

- Solution Installation and Deployment Dependency Checker.
- Managed Resource Touchpoints:
 - Generic Log Adapter Touchpoints.
 - Solution Installation and Deployment Touchpoints.
- Customization Tools:
 - Resource Model Builder.
 - Adapter Rule Editor.
 - Integrated Solutions Console Toolkit.
 - SMD Editor.
- User Access:
 - Integrated Solutions Console.

Se proveen varios ejemplos de implementaciones de un administrador autónomo. *AME* incluye capacidades incorporadas para las cuatro partes del ciclo de control del administrador autónomo (monitoreo, análisis, planificación y ejecución).

El *Log and Trace Analyzer* es un ejemplo de una implementación parcial del administrador autónomo, incluyendo a las partes de monitoreo y análisis del ciclo de control (control loop). La solución de la instalación y la tecnología de despliegue que son parte del *AC Toolkit* muestran otro ejemplo de un administrador autónomo.

La *Dependency Checker* ejecuta la función de monitoreo de un administrador autónomo. Se suministran algunas tecnologías y herramientas para ayudar a crear *touchpoints* (puntos de contacto) a los desarrolladores, los que permiten a los recursos administrados comunicarse con el administrador autónomo.

El *Generic Log Adapter* es un ejemplo de tal tecnología y se incluye en el *AC Toolkit* para traducir mensajes de log de productos al formato de datos de la *Common Base Event*.

El *AC Toolkit* también contiene herramientas para permitir personalizar implementaciones de *touchpoint* a un administrador autónomo y un recurso administrado.

El *RMB* se usa para personalizar un *AME*.

El *Adapter Rule Edition* se usa con *Generic Log Adapter*.

El *SMD Editor* ayuda a personalizar la instalación de la solución y las tecnologías de despliegue.

La *Integrated Solution Console Toolkit* permite construir plug-ins para la *Integrated Solutions Console*. El componente *Integrated Solution Console* provee el acceso a los usuarios a las capacidades de auto-administración.

Esta es una infraestructura basada en Web basado en tecnologías estándares de la industria (de la disciplina).

3.4 AME

3.4.1 Introducción

El *AC Toolkit* incluye *AME*, un ejemplo de una implementación de un administrador autónomo.

AME monitorea los recursos del sistema, envía eventos agregados, y desempeña acciones correctivas para los problemas. *AME* constantemente monitorea el sistema en busca de eventos para administrarlos.

AME está disponible en el *AC Toolkit* en el paquete *AME*. Un escenario de demostración *AME* está incluido en el *AC Toolkit*. El escenario muestra cómo *AME* puede monitorear una situación, detectar la situación, y proveer una acción correctiva.

3.4.2 Modelo de Recursos AME

Definiendo un *modelo de recurso* por cada recurso administrado, se provee a *AME* con el conocimiento requerido para administrar esos recursos. Los modelos de recursos contienen métricas específicas, eventos, umbrales, y parámetros, los cuáles son usados para determinar la vitalidad de los recursos junto con especificaciones para acciones correctivas en el caso de fallas.

AME provee servicios para instalar, arrancar, y parar un modelo de re-

curso, y averigua su estado. El *AC Toolkit* provee una herramienta que puede usarse para crear un modelo de recurso *AME* propio del usuario, el *RMB*. Un ejemplo de modelo de recurso *AME* se proporciona en el *AC Toolkit* para propósitos educativos y de demostración. Cualquiera puede utilizar esta muestra para aprender cómo se escriben los modelos de recursos de *AME* y cómo quedan incluidos en el *AME*.

3.5 **Log and Trace Analyzer**

El *Log and Trace Analyzer* (LTA) es un ejemplo de una implementación parcial del administrador autonómico, cubriendo las partes de análisis y monitoreo del ciclo de control. El LTA permite la visión, el análisis, y la correlación de archivos log. Esta herramienta lo hace más fácil y rápido para eliminar errores y para resolver problemas dentro de múltiples sistemas utilizando datos en el formato *Common Base Event* y proporcionando la visualización y el análisis especializados de los datos.

El *LTA* contiene un motor de análisis de log. La función de este motor es proporcionar un algoritmo que toma un incidente que se registra en un archivo log como un parámetro de entrada, busca coincidencias entre este incidente basado en las reglas predefinidas y los síntomas de una base de datos de síntomas disponible y retorna un vector de objetos que representan las soluciones y las directivas para los síntomas coincidentes.

El *LTA* provee una implementación por defecto de un motor de análisis y un conjunto de instrumentos que podrían usarse para implementar un motor de análisis del cliente.

El *AC Toolkit* contiene un motor de correlación por defecto como parte del paquete del *Log and Trace Analyzer*. Las capacidades incluyen el timestamp y la correlación de registro ID. También se proporciona la capacidad de crear el motor de correlación del cliente. El LTA está disponible en el *AC Toolkit* en el paquete de *GLA/LTA*.

Product analizador de log: El *AC Toolkit* provee un programa analizador para varios productos de IBM. Ellos están incluidos en el paquete *GLA/LTA*. Estos programas analizadores junto con los desarrollados por los propios productos de log del cliente, podría usarse para depurar problemas complejos, como el desarrollo de aplicaciones de auto-administración en un entorno de

sistema de multiproducto.

3.6 Instalación de la Solución y Tecnologías de Despliegue

La instalación de la solución y las tecnologías de despliegue son también parte del *AC Toolkit*.

A través de escenarios basados en instalación de aplicaciones de software de líderes de la industria, el *AC Toolkit* muestra cómo éstas tecnologías pueden usarse para mejorar las tareas de instalación y despliegue. Una tecnología que se introduce en esta versión está en el área de chequeo de dependencia.

Chequeador de Dependencia: Este es otro ejemplo de una implementación parcial de un administrador autonómico. El chequeador de dependencia implementa la parte de *análisis* de un administrador autonómico. El chequeador de dependencia es llamado por el código de instalación de la aplicación para validar las dependencias codificadas en un *Solucion Module Descriptor* (SMD).

Regresa una respuesta verdadera o falsa basado tanto en la existencia o no de dependencia. Este está al nivel del código de instalación de la aplicación para determinar la acción de esa respuesta. Las acciones apropiadas podrían ser, no hacer caso de la condición y continuar la instalación, registrar un mensaje, exhibir un panel al usuario, o terminar el proceso de la instalación.

Editor SMD: El *Editor SMD* es una herramienta usada para crear XML que describe la unidad instalable (IU) y sus dependencias asociadas. Este XML provee el conocimiento que el chequeador de dependencia usa para analizar sistemas de dependencias efectivamente.

Se puede editar un *SMD* existente o empezar de cero.

3.7 Adaptador Genérico de Log

El *Generic Log Adapter* (GLA) o *Adaptador Genérico de Log:* es un ejemplo de una tecnología que ayuda a un producto a crear un modelo de *touchpoint* de un recurso AC. El *GLA* provee la habilidad de tomar un archivo log de un producto de IBM o no, y convertir los mensajes al formato de datos *Common*

Base Event, así que el producto puede convertirse en un recurso administrado.

El *GLA* traduce las entradas de log de productos al *Common Base Events* para uso de un administrador autonómico. El *AC Toolkit* incluye el *GLA* como una tecnología para ayudar a los productos a adaptarse a la *Arquitectura de Referencia Autonómica* sin requerir que el producto cambie la manera de crear los archivos log.

Un run time (software de tiempo de ejecución) *GLA* sencillo puede usarse para analizar los archivos log de múltiples productos mientras las reglas han sido definidas para cada formato de mensaje. El adaptador incluye un manipulador que pasa la información *Common Base Event* al administrador autonómico mediante la interfase de manejabilidad.

El *GLA* está disponible en el *AC Toolkit* en el paquete *GLA/LTA*. Un escenario que muestra el *GLA* en una aplicación de auto-reparación se incluye en el *AC Toolkit*.

En este escenario, *GLA* lee archivos log de productos actuales en tiempo real. Usando un conjunto de reglas del programa analizador suministrado para los productos del escenario, el adaptador traduce cada mensaje log al formato *Common Base Event*. El *GLA* se muestra en el paquete de *Problem Determination Scenario*.

Adapter Rule Editor o *Editor de Reglas del Adaptador*: Esta herramienta se usa en conjunto con el *GLA*. Provee la herramienta para crear las reglas de análisis específicas que son usadas por el *GLA* en tiempo de ejecución para crear objetos *Common Base Event*. El *Adapter Rule Editor* está disponible en el *AC Toolkit* en el paquete *GLA/LTA*.

3.8 Consola de Soluciones Integradas

Integrated Solutions Console o *Consola de Soluciones Integradas*: El objetivo central de la *Consola de Soluciones Integradas* (ISC) es crear una plataforma sobre la que los productos IBM o no-IBM puedan construir interfases de usuario administrativas.

Estandarizar los productos para ejecutar sobre la plataforma *ISC* les da una apariencia y sensación más común y un comportamiento más constante, porque se usan bloques de construcción comunes.

Los administradores pueden interactuar con múltiples productos IBM o no-IBM desde una consola basada en navegación.

ISC se basa en el *WebSphere Portal*, así las funciones administrativas son manipuladas a través de *portlets*, o componentes, dentro de un sistema único. Cuando un administrador agrega nuevo software, sus funciones administrativas y archivos de ayuda son agregados al sistema administrativo común.

El *ISC Toolkit* es el entorno de desarrollo para crear plug-ins *ISC* y se incluye en el paquete *ISC*. Incluye el run time (tiempo de ejecución) *ISC*, el *Integrated Solutions Console Developer Info Center* o *Centro de Información de Desarrollo de Consolas de Soluciones Integradas* y un plug-in *ISC* para *WebSphere*.

Un componente de plug-in específico de *ISC* se provee en el *AC Toolkit* para soportar el *Problem Determination Scenario* o *Escenario de Determinación de Problema*.

La consola de *AC Toolkit* se configura para responder a los scripts de modelo de recurso AME. Estos scripts ajustan los parámetros de estado según van sucediendo situaciones, manteniendo informados a los administradores sobre la condición de sus productos.

Además funciona como un ejemplo de cómo podrían ser creados los plug-ins para otras aplicaciones de administración de productos de usuario específicos. La consola *AC Toolkit* es solamente un ejemplo instructivo e ilustra sólo una metodología posible. El componente *ISC* provee un completo conjunto de documentación para el run time (tiempo de ejecución) así como también para el *ISC Toolkit*.

Capítulo 4

Escenarios e Instalación de AC Toolkit

4.1 Objetivos

Uno de los principales objetivos del *AC Toolkit* es proveer una muestra, fácil de entender, de cómo las tecnologías del *AC Toolkit* pueden usarse para resolver problemas del mundo real. Éstas muestras se proveen como una colección de escenarios que se enfocan sobre atributos específicos de un entorno de AC.

Los tres escenarios incluidos en el *AC Toolkit* son simples representaciones de típicos puntos de preocupación del cliente que puede gestionarse usando tecnologías AC. El propósito de éstos escenarios es demostrar cómo trabajan los componentes del *AC Toolkit* en conjunto con el objeto de resolver problemas del mundo real. Estos escenarios se sustentan con documentación y pueden ser usados como ejemplos para ayudar a desarrollar una solución autónoma [4].

4.2 Escenario de Determinación de Problema

El escenario de Determinación de Problema muestra el atributo de auto-administración que el AC aporta a un entorno informático. Para que un sistema sea auto-reparable, necesita ser capaz de reconocer que ha ocurrido un problema, determinar la causa, y entonces tomar la acción adecuada para

corregir el problema. Un método de lograr esto es a través de archivos log existente de un producto. Una de las tecnologías de *AC Toolkit* usada en el escenario es el *GLA* que convierte eventos de archivos de log de productos al formato de datos Common Base Event.

El escenario de Determinación del Problema muestra cómo un administrador autónomo, representado aquí por *AME*, puede ser usado para detectar una situación de error entre dos productos IBM analizando los archivos de log/trace (seguimiento/rastreo) y aplicando una acción correctiva.

El escenario usa dos productos IBM interactuando mutuamente para mostrar cómo se detecta y resuelve un problema común basado en Web. Se incluye un producto sencillo de base de datos para ejecutar consultas básicas con acceso a bases de datos. La pregunta se origina desde una aplicación Web, también incluida en el paquete del escenario.

El *ISC* es otra tecnología del *AC Toolkit* que se usa en este escenario. El escenario incluye una consola de administración construida con la tecnología *ISC*. Éste es usado para arrancar y terminar el escenario y además provee un método de inducción de condición de falla y capacidades de monitoreo de estado. La consola de administración arranca el escenario en un estado estable mostrando la operación normal de los dos productos. Entonces, éste permite inducir una falla de base de datos que es reconocida desde los archivos log de las aplicaciones Web.

El modelo de recursos *AME* incluidos se programa para identificar la falla de la base de datos. Después de que la situación haya sido detectada por *AME*, la acción correctiva es emitida al recurso administrado. Cada fase de la operación del ciclo de control se muestra en el panel de estados de la consola administrativa. Después de que haya sido corregida, la aplicación Web comienza a funcionar otra vez y el escenario regresa al estado estable.

El *escenario de Determinación del Problema* ilustra el uso de las tecnologías en el *AC Toolkit* para llevar a cabo las siguientes tareas:

- Transformar un producto de archivo log en un formato de datos Common Base Event.
- Personalizar *AME* con modelos de recursos.
- Mantener la comunicación entre el administrador autónomo y los recursos administrados.

- Construir una consola administrativa usando la tecnología ISC.
- Construir un sistema funcional sencillo capaz de auto administrarse.

Después de ejecutar y observar el escenario de Determinación del Problema, se puede aplicar la misma técnica para los propios productos y diseñar las soluciones propias para la determinación de un problema.

4.3 Escenario de Instalación Automática

Los atributos de auto-configuración pueden mostrarse usando la solución de instalación y tecnologías de despliegue disponibles en el *AC Toolkit*.

Los beneficios y versatilidades de la solución de instalación y tecnologías de despliegue autonómicas se muestran proporcionando dos escenarios usando dos diferentes productos de software de instalación proveídos por diferentes fabricantes.

El propósito de los escenarios es demostrar los conceptos que respaldan a la solución de instalación y tecnologías de despliegue mostrando la *instalación de un paquete de software basado el Web*.

Cada escenario consiste de código, junto con un paquete descriptor que explica el contenido y pre-requisitos. Éste paquete descriptor se llama “*unidad instalable*” (UI o IU en inglés), y múltiples unidades son agrupadas dentro de módulos de la solución.

La instalación de software de aplicación lee el archivo descriptor ejecutando la instalación actual y chequea una base de datos de software y hardware instalado para determinar si todos los pre-requisitos han sido encontrados. En caso de que se disponga de ellos, el software es instalado y su información es agregada a la base de datos. Si no fuera así, el instalador corrige el problema localizando el software requerido e instalándolo para satisfacer los requisitos necesarios.

Se proveen dos escenarios basados en dos generadores de instaladores de aplicaciones de software. Un escenario usa *ISMP* desde el *InstallShield* como el software generador, mientras que el otro escenario usa *InstallAnywhere* desde Zero G. El escenario generador de instaladores contiene el run time (tiempo

de ejecución) de la solución de instalación y de despliegue, así como también el necesario autoarranque.

4.4 El Paquete AC Toolkit

El contenido del *AC Toolkit* es muy amplio y variado para ser descargado íntegramente; por lo tanto, gran cantidad de contenido había estado sin ser aprovechado dentro de un puñado de paquetes fáciles de obtener. Este paquete ayuda a tomar el contenido correcto sin tener que conocer todos los detalles específicos del AC Toolkit y se evita bajar más contenido del que uno quiere o necesita.

Los siguientes paquetes están disponibles en el AC Toolkit:

- Autonomic Computing Information.
- AME.
- RMB.
- Integrated Solutions Console.
- Solution Installation and Deployment.
- Generic Log Adapter and Log and Trace Analyzer.
- Problem Determination Scenario.
- Solution Installation and Deployment Scenario Using ISMP.
- Solution Installation and Deployment Scenario Using InstallAnywhere.

Las siguientes secciones describen, en detalle, cada uno de los paquetes disponibles.

4.4.1 AC Information

Este paquete contiene todo lo pertinente a las referencias de AC, documentación, y tutoriales que describen al *AC Toolkit*. Es un buen punto de comienzo para aquellos que quieren llegar a saber más sobre AC y *AC Toolkit*:

- *Contenido del Paquete:* Este paquete incluye sólo documentación.
- *Pre requisitos:* Este paquete no tiene pre requisitos.
- *Paquetes Relacionados:* Esta documentación describe el *AC Toolkit* de manera general.

4.4.2 AME

Este paquete contiene el componente *AME*:

- *Contenido del Paquete:* Este paquete incluye:
 - AME V1.0.
 - Documentación (AME 1.0 Developer's Guide).
- *Pre requisitos:* Este paquete no tiene pre requisitos.
- *Paquetes relacionados:* Los siguientes paquetes están relacionados:
 - Problem Determination Scenario: Muestra una solución autonómica actual e incluye las API's estandarizadas.
 - RMB: Requerido si se desea modificar las funciones del AME.

4.4.3 RMB

Este paquete contiene el componente *RMB*.

- *Contenido del Paquete:* Este paquete incluye:
 - RMB V1.2.
 - Requiere el Workbench V2.1.2 basado en "Eclipse".
- *Pre requisitos:* Este paquete no tiene pre requisitos.
- *Paquetes relacionados:* Los siguientes paquetes están relacionados:
 - AME.
 - Problem Determination Scenario.

4.4.4 Integrated Solutions Console

Este paquete contiene el componente de *ISC*.

- *Contenido del Paquete:* Este paquete incluye:
 - Integrated Solutions Console run time V5.0.1.
 - Integrated Solutions Console V5.0.1.
 - Eclipse V2.1.1.
 - EMF V1.1.0.
- *Pre requisitos:* *ISC* requiere que el sistema tenga una entrada DNS y que pueda determinarse el nombre del host. Si el sistema no tiene una entrada DNS, se puede actualizar la siguiente en el archivo de hosts: **127.0.0.1.nombre.del.servidor**, donde **nombre.del.servidor** es el nombre del servidor si no se tiene una entrada DNS para el servidor.
- *Paquetes relacionados:* El paquete Problem Determination Scenario muestra cómo puede crearse y usarse un plug-in de *ISC* para administrar una real solución de auto-administración.

4.4.5 Solution Installation and Deployment

Este paquete contiene la solución de instalación y tecnologías de despliegue.

- *Contenido del Paquete:* Este paquete incluye:
 - Solution Installation and Deployment Run Time V1.1.
 - SMD Editor V1.0.
 - Documentación sobre cómo usar Solution Installation and Deployment para desplegar sistemas simples y complejos:
 - * Solution Installation and Deployment Technology Overview v1.1.
 - * Solution Installation and Deployment Developer's Guide v1.1.
 - * Solution Installation and Deployment Release Notes v1.1.
- *Pre requisitos:* *SMD Editor* requiere *Eclipse* y *EMF*.

- *Paquetes relacionados*: Los siguientes paquetes están relacionados:
 - Solution Installation and Deployment Scenario using ISMP.
 - Solution Installation and Deployment Scenario using Zero G.

4.4.6 Generic Log Adapter and Log Trace Analyzer

Este paquete contiene el *Generic Log Adapter* y el *Log and Trace Analyzer Tool*.

- *Contenido del Paquete*: Este paquete incluye:
 - Generic Log Adapter Run Time V1.3.
 - Rule Editor V1.3.
 - Product Rules V1.3.
 - Log and Trace Analyzer para AC V1.3.
 - Eclipse V2.1.1.
 - EMF V1.1.0.
- *Pre requisitos*: JDK 1.3.1 o superior se requiere para usar este paquete.
- *Paquetes relacionados*: El paquete *Problem Determination Scenario* muestra cómo el *Generic Log Adapter* es usado en tiempo real para una solución autónoma.

4.4.7 Problem Determination Scenario

Este paquete contiene el *Problem Determination Scenario*.

- *Contenido del Paquete*: Este paquete incluye:
 - Modelo de recurso Canonical Situation Monitor.
 - Integrated Solutions Console plug-in for Problem Determination Scenario.
 - Script para automatización para Problem Determination Scenario.

- Ejemplo de una aplicación WebSphere Application Server.
 - Configuración de archivos GLA-LTA para Problem Determination Scenario.
 - Base de Datos Cloudscape.
 - Clases Java de ayuda para la interfase de manejabilidad.
 - IBM Autonomic Computing Toolkit Problem Determination Log/Trace Scenario Guide.
-
- *Pre requisitos Importantes:* Los pre requisitos enumerados a continuación se requieren para la instalación de una funcionalidad del escenario de determinación de problemas. No obstante, una nueva opción de instalación permite dejar de lado la instalación de los pre requisitos para poder ir viendo los archivos del escenario. Esta instalación permite la extracción y la vista de los archivos pero no es una instalación de escenario funcional. Para cambios posteriores en una instalación funcional, se requiere una desinstalación y una reinstalación del paquete PDSscenario.
 - Se deben tener los siguientes componentes instalados para poder usar este paquete:
 - AME V1.0.
 - Integrated Solutions Console V5.0.1.
 - Generic Log Adapter and Log and Trace Analyzer for Autonomic Computing V1.3.
 - El modelo de recurso proporcionado con el Problem Determination Scenario requiere de un archivo fuente abierto llamado Regular Expression for Java gnu-regexp.jar v1.1.1 o superior.
-
- *Paquetes Relacionados:* Este paquete usa tecnologías contenidas en los siguientes paquetes:
 - Paquete AME.
 - Paquete RMB.
 - Paquete GLA y LTA.

4.4.8 Solution Installation and Deployment Scenario Usando ISMP

Este paquete contiene el escenario de la solución de instalación y de despliegue usando *ISMP*.

- *Contenido del Paquete:* Este paquete incluye:
 - Solution Installation and Deployment Run Time V.1.1.
 - InstallShield ISMP V5.0.
 - ISMP V5.0 Run Time.
 - Descriptor XML.
 - Solution Installation and Deployment Using ISMP.
- *Pre requisitos:* El Editor SMD requiere Eclipse y EMF.
- *Paquetes Relacionados:* Los siguientes paquetes están relacionados:
 - Solution Installation and Deployment Scenario.
 - Solution Installation and Deployment Scenario using InstallAnywhere.

4.4.9 Solution Installation and Deployment Scenario Using InstallAnywhere

Este paquete contiene la instalación de la solución y el escenario de despliegue usando *InstallAnywhere*.

- *Contenido del Paquete:* Este paquete incluye:
 - Solution Installation and Deployment Run Time.
 - Zero G InstallAnywhere.
 - Descriptor XML.
 - *Solution Installation and Deployment Using InstallAnywhere.*
- *Pre requisitos:* SMD Editor requiere Eclipse y EMF.

- *Paquetes relacionados*: Los siguientes paquetes están relacionados:
 - Solution Installation and Deployment Scenario.
 - Solution Installation and Deployment Scenario using ISMP.

4.5 Descarga del Paquete AC Toolkit

Cuando se está listo para probar una herramienta, tecnología, o escenario del AC Toolkit, se necesitará decidir qué paquete se requiere, se deberá asegurar que se cuenta con los pre requisitos necesarios, y descargar el código desde el sitio Web **developerWorks**: www.ibm.com/developerworks/autonomic/

Para descargar el código, se deben ejecutar los siguientes pasos:

1. Ir a la página principal del AC Toolkit. Esta página ayuda a decidir qué paquete o paquetes se desea descargar.
2. Leer la lista de pre requisitos y asegurarse de que éstos están instalados en el sistema.
3. Cuando todo esté listo para la descarga, ir a la página de descargas.
4. Hacer click sobre el link del paquete seleccionado.

Si existiera algún inconveniente con este proceso, enviar una pregunta al foro de ayuda. La primera vez que se usa el foro se debe crear una identificación de usuario y una clave, pero se debe volver a usar esta misma identificación cada vez que se envía una pregunta al foro.

4.6 Instalación del Paquete AC Toolkit

Esta sección describe la instalación del paquete *AC Toolkit*.

4.6.1 Vista General de la Instalación

Los paquetes del *AC Toolkit* generalmente son instalados de la misma manera para mayor consistencia. Así mismo, algunos paquetes contienen sólo un com-

ponente, mientras que otros contienen múltiples componentes, lo cuál implica que la instalación se presente de diferentes modos.

Los paquetes son archivos ejecutables que ejecutan sus propias instalaciones; esto ayuda a crear instalaciones consistentes y minimiza errores de usuario. La información de la registración es logeada dentro del registro cuando se instala el paquete. Esta información registrada es usada para futuras instalaciones y chequeos de pre requisitos.

4.6.2 Prerequisitos

Los siguientes ítems son requeridos para correr el *AC Toolkit*:

- Procesador Intel Pentium II (Pentium III 500 MHz o superior recomendado).
- 512 MB de RAM (768 MB de RAM recomendado).
- Espacio en disco requerido:
 - Espacio en disco para instalación como se muestra en la Tabla 4.1 de la pág. 50 y en la Tabla 4.2 de la pág. 51.
 - Espacio en disco adicional para el despliegue de recursos.
 - Espacio en disco adicional requerido si se descarga la imagen electrónica para la instalación de esta tecnología¹.
- Monitor con resolución de 800 x 600 (1024 x 768 recomendado).
- Uno de los siguientes navegadores Web para uso del ISC y la visualización, tanto de las licencia de uso como de la ayuda en línea:
 - Para Linux y sistemas AIX, Netscape 7.
 - Para sistemas Windows:
 - * Microsoft Internet Explorer 6.x y 7.
 - * Mozilla 1.0.2.

¹EL mínimo espacio en disco puede reducirse si las características opcionales y los entornos de run time (tiempo de ejecución) no son instalados.

- Java con el path del sistema actualizado y la variable `JAVA_HOME`, actualizada. Las siguientes JVMs son soportadas:
 - IBM Java 2 Standard Edition V1.4.1 JRE or Developer Kit.
 - Sun Java 2 Platform, Standard Edition (J2SE) V1.4.1 (J2SE) JRE or SDK.

Componentes	Espacio requerido
AME	12 MB
GLA/LTA	235 MB
Integrated Solution Console	224 MB
Problem Determination Scenario	27 MB
Solution Installation and Deployment InstallAnywhere Scenario	228 MB
Solution Installation and Deployment ISMP Scenario	316 MB
Total	1024 MB

Tabla 4.1: Espacio requerido por instalaciones Linux y AIX.

4.6.3 Plataformas Soportadas

Todos los componentes del *AC Toolkit* requieren una de las siguientes plataformas:

- Componentes run time (tiempo de ejecución) y escenarios:
 - IBM AIX V5.2.
 - Red Hat Enterprise Linux V2.1 (en Intel 32).
 - SuSe Linux Enterprise Server V8 (en Intel 32).
 - Microsoft Windows 2000 Server (Service Pack 3 o superior).
 - Microsoft Windows 2000 Advanced Server (Service Pack 3 o superior).
- Facilidades:

Componentes	Espacio requerido
AME	12 MB
GLA/LTA	235 MB
Integrated Solution Console	235 MB
RMB	55 MB
Problem Determination Scenario	27 MB
Solution Installation and Deployment Development	25 MB
Solution Installation and Deployment InstallAnywhere Scenario	228 MB
Solution Installation and Deployment ISMP Scenario	315 MB
Total	1132 MB

Tabla 4.2: Espacio requerido por instalaciones Windows.

- Microsoft Windows 2000 Server (Service Pack 3 o superior).
- Microsoft Windows 2000 Advanced Server (Service Pack 3 o superior).
- Microsoft Windows XP (Service Pack 1).

4.6.4 Instalación del Paquete AC Toolkit

Para instalar el paquete *AC Toolkit*, se deben realizar los siguientes pasos:

1. Verificar los requerimientos de sistema necesarios para ejecutar el paquete.
2. Doble cliquear en el archivo ejecutable para comenzar la instalación. Aparece el panel Welcome.
3. Cliquear Next. Si el paquete ya está instalado en el sistema, un mensaje de advertencia aparecerá indicando que una versión previa está instalada y pregunta si se desea proseguir. Para proseguir, realizar uno de los siguientes pasos:
 - Cliquear Cancel en la ventana del mensaje de advertencia para abortar la nueva instalación y mantener la instalación previa.

- Clicar Yes para continuar la instalación.
4. Aceptar los términos del panel de Licencia y clicar Next. El panel Install Location muestra la ubicación por defecto de la instalación. El panel Feature List lista los detalles de la instalación.
 5. Realizar una de las siguientes acciones: aceptar la ubicación por defecto o reemplazarla por otra. Clicar Next para instalar en esa ubicación.
 6. Cuando se completa la instalación, una nueva ventana de mensaje informa que el proceso ha finalizado. Clicar Finish para salir del asistente de instalación².

4.6.5 Desinstalación del Paquete AC Toolkit

Para desinstalar el paquete *AC Toolkit*, realizar los siguientes pasos:

1. Doble clic en el archivo ejecutable Uninstall.exe situado en el directorio de instalación. Aparece el panel de Welcome.
2. Clicar Next. Se muestran detalles de la desinstalación.
3. Clicar Next para desinstalar el paquete de la máquina. Cuando se completa la desinstalación, aparece una ventana de mensaje final.
4. Clicar Finish para salir del asistente de desinstalación.

4.7 Instalación de la ISC

Esta sección describe cómo instalar *ISC*.

4.7.1 Antes de Comenzar

Antes de comenzar el proceso de instalación de la *ISC*, de deben ejecutar los siguientes pasos:

²Se recomienda que se ejecute una instalación a la vez.

1. Desinstalar cualquier versión anterior de la *ISC*.
2. Opcionalmente, borrar todos los directorios temporales antes de proceder con la instalación de la *Integrated Solutions Console*.

4.7.2 Requerimientos Para la Instalación de la ISC

1. Previo a la instalación de este escenario debe ser instalado uno de los siguientes paquetes *JRE 1.4.1*:
 - IBM Java 2 Standard Edition V1.4.1 JRE o Developer Kit.
 - Plataforma Sun Java 2, Standard Edition (J2SE) V1.4.1 (J2SE) JRE o SDK2.
2. La instalación del *ISC* requiere como mínimo 275 MB de espacio libre en la unidad %tmp% en *Windows* y 550 MB de espacio libre en la unidad \$tmp para *AIX* y *Linux*.
3. Chequeo de una dirección IP estática.

La *ISC* asume que el sistema de host está usando una dirección IP estática en lugar de una dirección IP asignada dinámicamente. Por lo tanto, se debe configurar la máquina usando una dirección IP estática. Si se instala la *ISC* para el uso de sólo una persona y ningún otro usuario necesitará acceder a la instalación *ISC*, se debe configurar el sistema a fin de que el puerto IP se mapee al nombre de host cualificado completamente.

Para facilitar el mapeo, se agrega las siguientes líneas al archivo de hosts. Por ejemplo, en *Windows* el archivo de hosts se localiza en *C:\WINNT\system32\drivers\etc\host*:

```
127.0.0.1 localhost
127.0.0.1 nombre.del.servidor.
```

1. Chequear los múltiples adaptadores siguiendo los siguientes pasos:
 - En *Windows*, clicar **Settings** → **Control Panel**. Se despliega la ventana del Panel de Control. Clicar **Network and Dial-up Connections**. Se despliega la ventana de Conexiones de Redes y Dial-up.

- En el menú principal, clicar **Advanced** → **Advanced Settings**. Se despliega la ventana de Configuraciones Avanzadas.
 - En la lista **Connections**, verificar que el adaptador de red adecuado está primero en la lista. Si el adaptador no está primero, seleccionarlo y usar el teclado de flechas para moverlo al principio de la lista. Ejecutar la instalación después de finalizar este procedimiento.
2. Si se usa el *IBM JRE/JDK 1.4.1* como la *JVM* predeterminada, ejecutar los siguientes pasos antes de proceder con la instalación de la *ISC* para Windows; esto habilita al instalador de la *ISC* a usar el Java embebido con el paquete instalado *ISC*. Se debe observar que el directorio `%TEMP%\ISCImage` se crea durante la instalación y se borra después de la instalación.
 - (a) Abrir la ventana de comando.
 - (b) Tippear: `SET PATH=%TEMP%\ISCImage\RuntimeExt\ewase\windows\java\bin;%PATH%` y presionar **Enter**.
 - (c) Tippear `ISC_Win32.exe` y presionar **Enter** para comenzar la instalación de la *ISC*.
 - (d) Después de completarse la instalación, cerrar la ventana.
 3. El *Toolkit ISC* puede ser instalado sobre *Windows XP*, sin embargo, durante la instalación, se reporta un error que expresa que el *ISC runtime* no pudo ser instalado. Aparece un aviso para continuar.
 - Se selecciona **Continuar** para instalar sin el runtime.
 - Se selecciona **Cancel** para abandonar la instalación.

4.7.3 Instalación de la ISC

Para instalar la *ISC*, ejecutar los siguientes pasos:

La Fig. 4.1 de la pág. 55 muestra la ventana de *Bienvenida* que aparece en la pantalla con el nombre del producto y la corporación. Clicar **Next**.

La ventana de *Licencia* aparece mostrando la licencia. Ver Fig. 4.2 de la pág. 55.



Figura 4.1: Ventana de bienvenida a la ISC.

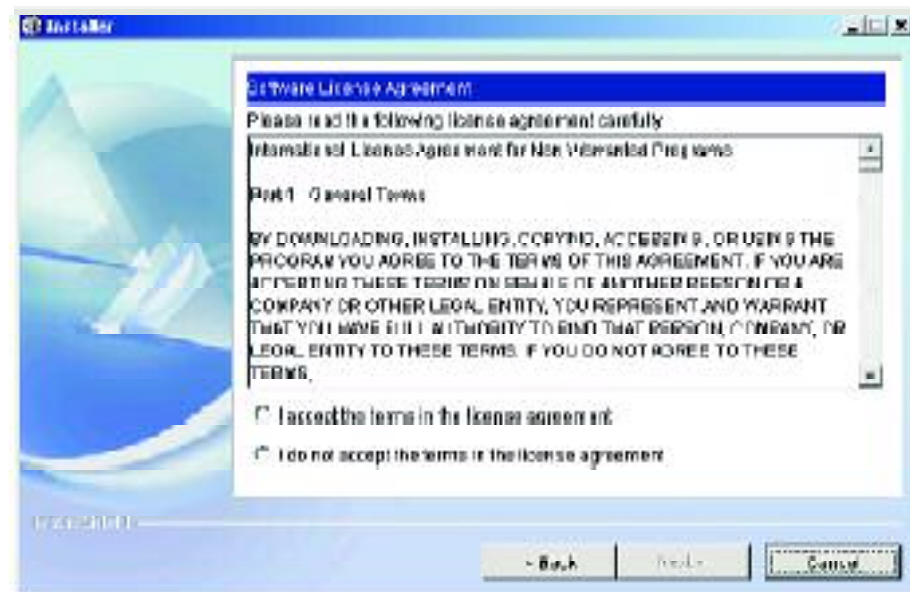


Figura 4.2: Ventana de licencia de la ISC.



Figura 4.3: Ventana de información de la ubicación de la instalación de la ISC.

Seleccionar el botón de elección: **I accept the terms in the license agreement** y clicar **Next**. Ver Fig. 4.2 de la pág. 4.2. Aparece la ventana de información de la instalación, la cual brinda información acerca de la ubicación de la instalación. Ver Fig. 4.3 de la pág. 56.

Ingresar la *ubicación* donde la *ISC* será instalada o aceptar la ubicación que se muestra como predeterminada. Clicar **Next** para continuar con la ventana de *Cuentas de Administrador de la ISC*. Ver Fig. 4.4 de la pág. 57.

Se debe ingresar el *nombre de usuario* del servidor del portal de la ISC y la *contraseña* como muestra en la Fig. 4.4 de la pág. 57. Este nombre de usuario y contraseña se requerirán para ingresar a la *consola*. Clicar **Next**.

Completar el nombre del *host* y clicar **Next**. Ver Fig. 4.5 de la pág. 57.

Completar la localización de los *puertos disponibles* o aceptar los predeterminados. Cuando se finaliza con la primer ventana (ver Fig. 4.6 de la pág. 58), clicar **Next** para pasar a la segunda ventana de puertos, también disponibles (ver Fig. 4.7 de la pág. 58).

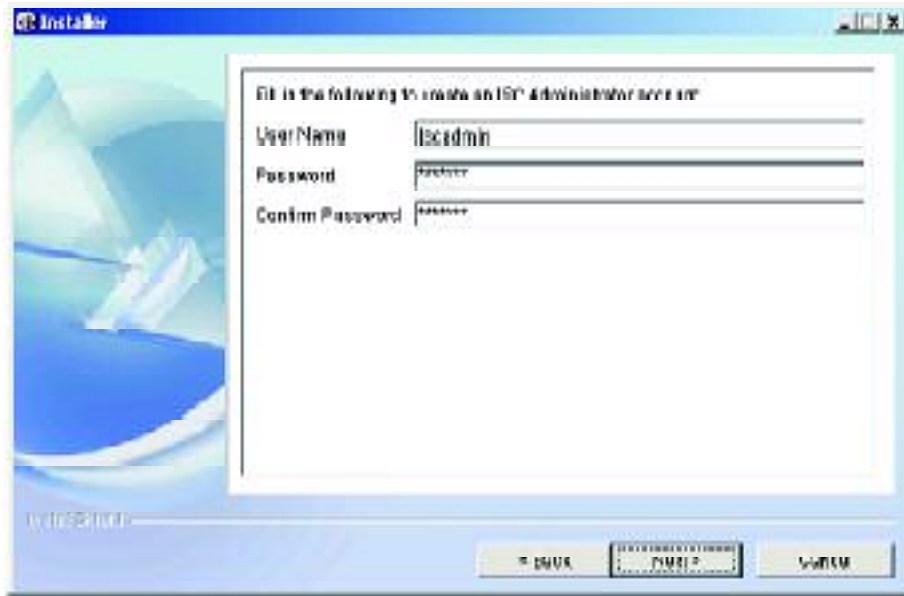


Figura 4.4: Ventana de cuentas de administrador de la ISC.



Figura 4.5: Nombre del host y localización del puerto.

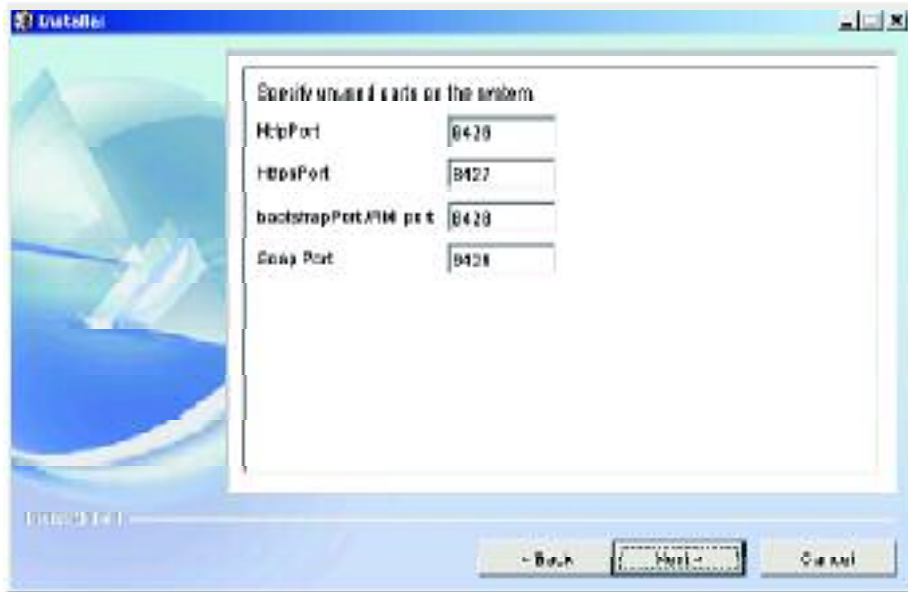


Figura 4.6: Localizaciones de puertos nuevos - ventana 1.

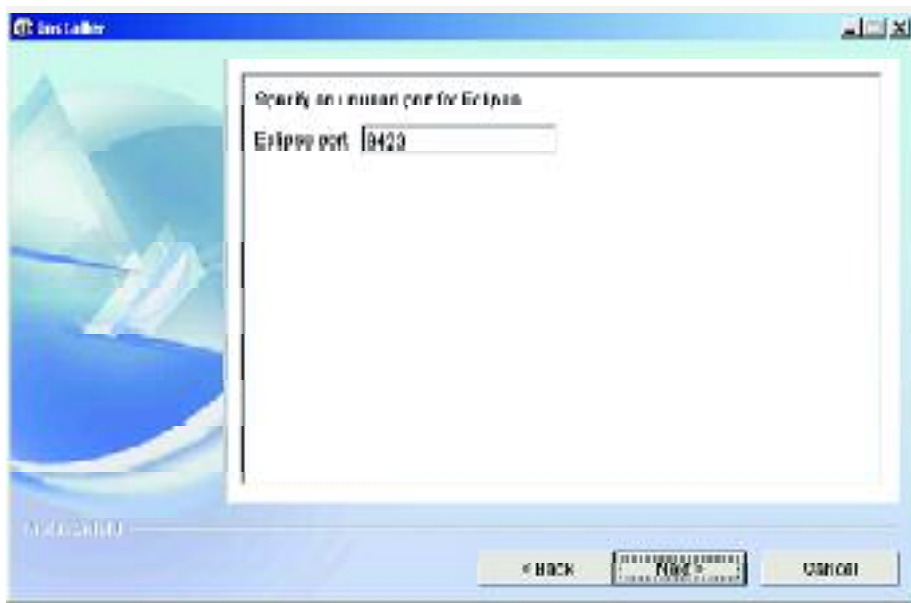


Figura 4.7: Localizaciones de puertos nuevos - ventana 2.

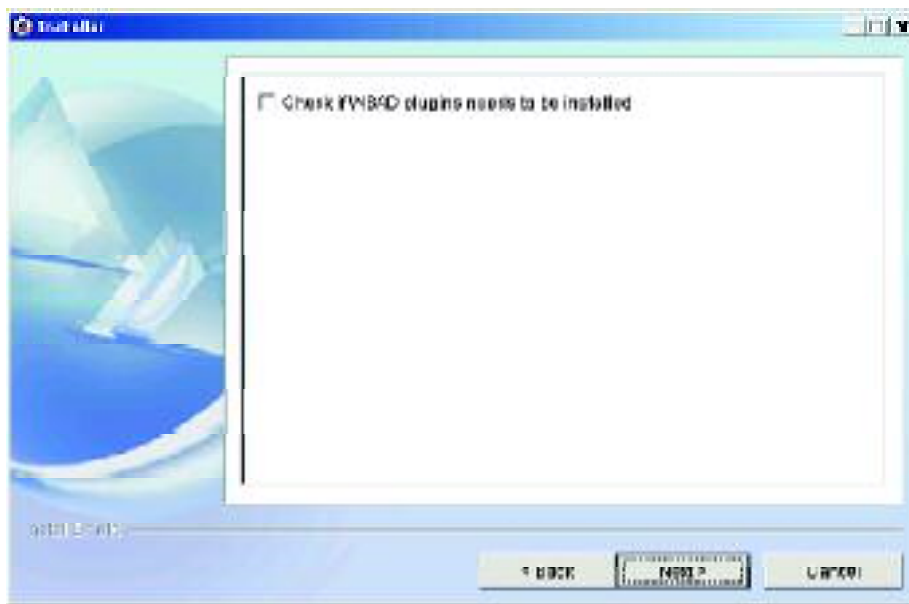


Figura 4.8: Ventana de plug-ins del WSphere Application Developer.

La ventana mostrada en la Fig. 4.8 de la pag. 59 ofrece la opción de instalar *WebSphere Studio Application Developer* plug-ins durante esta instalación. Seleccionar la caja de verificación si los plug-ins de *WebSphere Studio Application Developer* necesitan estar instalados. Clicar **Next**.

Si se selecciona la caja de verificación mostrada en la Fig. 4.8 de la pág. 59, aparece la ventana mostrada en la Fig. 4.9 de la pág. 60. Ingresar el directorio donde será instalado el *WebSphere Studio Application Developer* y clicar **Next**.

La ventana de *resumen de la ISC* muestra detalladamente la localización de la instalación, características seleccionadas, y el tamaño total de la instalación (ver Fig. 4.10 de la pág. 61). Verificar la información y clicar **Next**.

La Fig. 4.11 de la pág. 61 y la Fig. 4.12 de la pág. 62 muestran el porcentaje de terminación del proceso de instalación. Cuando el proceso se completa, clicar **Next**.

Cuando aparece la última ventana mostrada en la Fig. 4.13 de la pág. 62, clicar **Finish** (el proceso final de instalación puede tomar varios minutos).

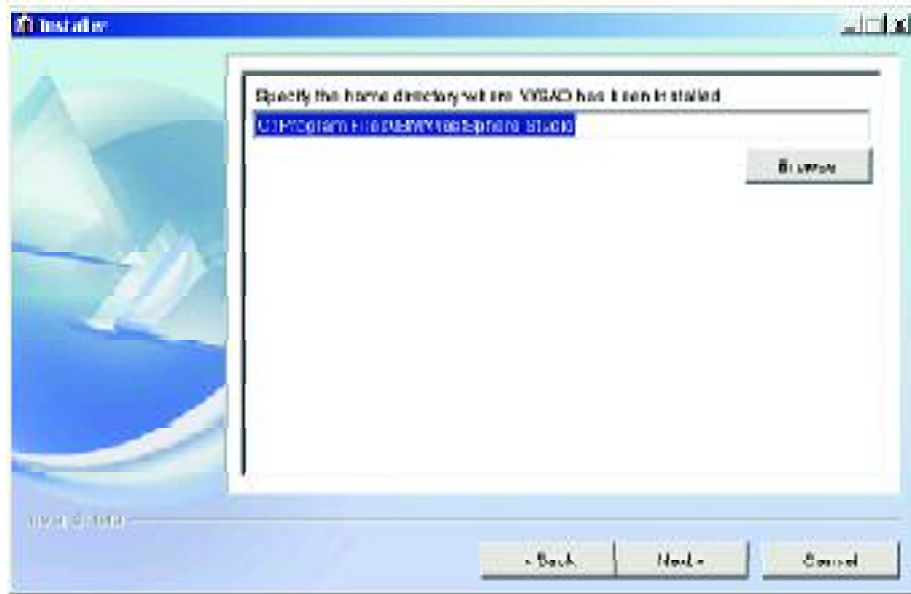


Figura 4.9: Ventana de localización de instalación del WebSphere Studio Application Developer.

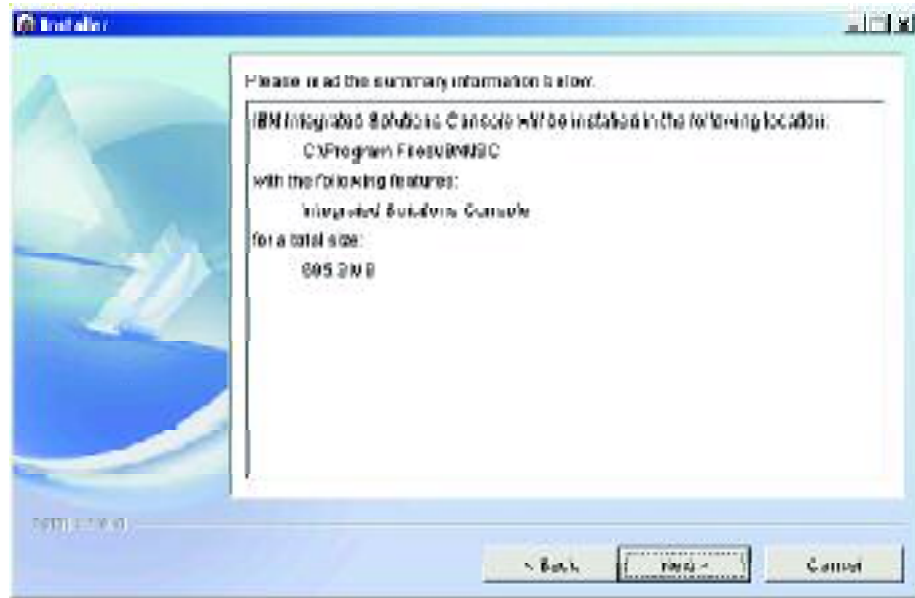


Figura 4.10: Ventana de localización e información de la instalación de la ISC.



Figura 4.11: Proceso de extracción de la ISC.

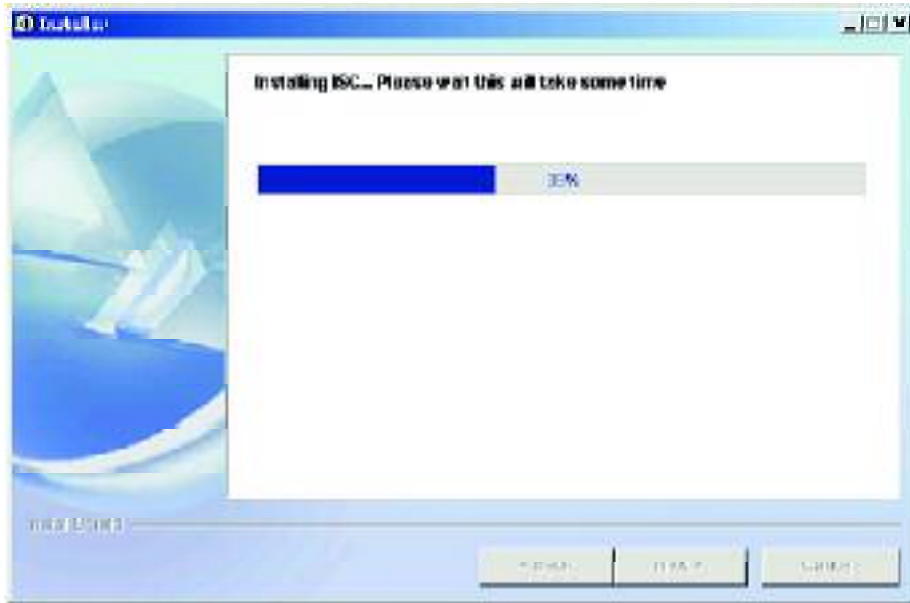


Figura 4.12: Proceso de finalización de la ISC.

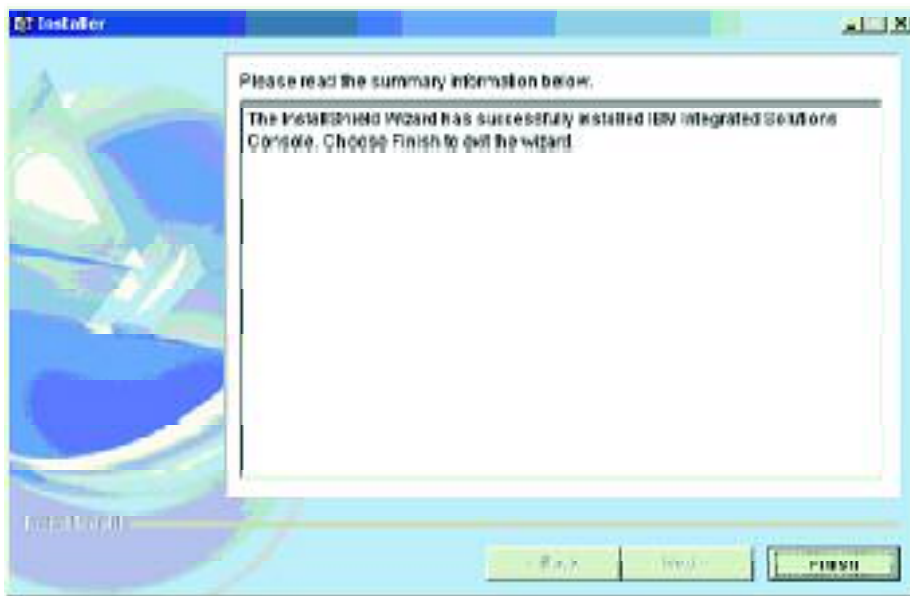


Figura 4.13: Ventana del proceso final de la instalación de la ISC.

4.7.4 Desinstalar la ISC

Para desinstalar la *ISC*, se deben ejecutar los siguientes pasos:

1. Para *Windows*:
 - (a) Ir a **Start** → **Settings** → **Control Panel**.
 - (b) Doble click sobre el ícono de **Add/Remove Programs**.
 - (c) Elegir **IBM Integrated Solutions Console** en la lista de aplicaciones y clicar el botón **Change/Remove** para desinstalar la *ISC*.
 - (d) Reiniciar el sistema.
2. Para *Linux* y *AIX*:
 - (a) Ir a la carpeta: `uninst2`.
 - (b) Doble click sobre el archivo: `uninstaller.bin`.
 - (c) Reiniciar el sistema.

4.8 Arrancar y Detener la ISC

Después de una instalación exitosa, el programa de instalación de la *ISC*, arranca automáticamente a la *ISC* y su sistema de ayuda. Cuando la máquina se reinicia, el *servidor del portal de la ISC* está en estado de detención. La *ISC* no funciona si el portal del servidor *ISC* no está activado.

Para *arrancar la ISC y su sistema de ayuda*:

1. Abrir la ventana de comando en el sistema donde está instalada la *ISC*.
2. Ejecutar los siguientes comandos para arrancar el servidor:
 - Para sistemas *Windows*, tipear: `la_raiz_isc \Runtime \AppServer \bin \startServer.bat ISC_Portal` y presionar **Enter**.
 - Para *AIX*, *Linux*, y sistemas *Solaris*, tipear: `la_raiz_isc /Runtime /AppServer /bin /startServer.sh ISC_Portal` y presionar **Enter**.

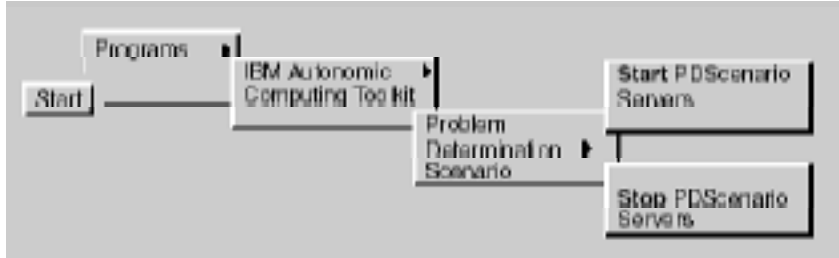


Figura 4.14: Windows.

Para cualquier sistema, *la_raiz_isc* es el directorio raíz de la instalación de la *ISC*.

3. Ejecutar los siguientes comandos para *arrancar el sistema de ayuda de la ISC*:

- Para sistemas *Windows*, tipear: `la_raiz_isc \Runtime \PortalServer \ISCEclipse \StartEclipse.bat` y presionar **Enter**.
- Para *AIX*, *Linux*, y sistemas *Solaris*, tipear: `la_raiz_isc /Runtime /PortalServer /ISCEclipse /StartEclipse.sh` y presionar **Enter**.

Para cualquier sistema, *la_raiz_isc* es el directorio raíz de la instalación de la *ISC*.

4.8.1 Secuencia Gráfica de Arranque y Detención de la ISC

En la Fig. 4.14 de la pág. 64, la Fig. 4.15 de la pág. 65 y la Fig. 4.16 de la pág. 65, se muestra gráficamente la secuencia de pasos necesaria para *arrancar* y *detener* la *ISC*.

Para *Windows*:

- El arranque directo está enlazado al archivo `startISCServers.bat` que abre `startServerISC_Portal@server1` desde la ubicación de la *ISC*.

Para *Linux*:

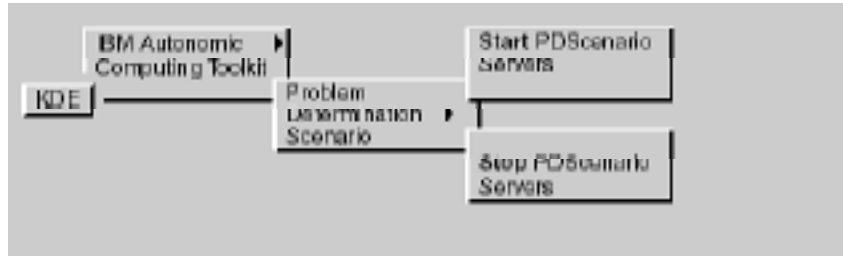


Figura 4.15: Linux.

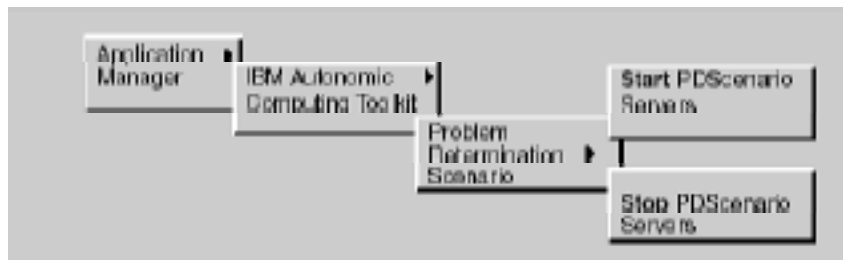


Figura 4.16: AIX.

- El arranque directo está enlazado al archivo *startiscervers.sh* que abre *ISC_Portal&server1* desde la ubicación de la *ISC*.

Para *AIX*:

- El arranque directo está enlazado al archivo *startiscervers.sh* que abre *ISC_Portal&server1* desde la ubicación de la *ISC*.

En *AIX*, los productos son instalados por el usuario raíz. Como resultado, los accesos directos solamente están accesibles para el usuario raíz. Si otros usuarios necesitan acceder al producto y a los accesos directos, se necesita cambiar manualmente los permisos. Después de que los accesos directos son agregados a cada usuario, se debe actualizar las aplicaciones de escritorio para ver el acceso directo en el administrador de aplicaciones ejecutando los siguientes pasos:

1. Abrir el **Application Manager** → **Desktop Tools**.
2. Doble click sobre: **Reload Desktop Applications**.
3. Regresar al Application Manger. Ahora está disponible el acceso directo (shortcut) del *IBM Autonomic Computing Toolkit*.

Para *detener la ISC y su sistema de ayuda*:

1. Abrir una ventana de comando en el sistema donde fue instalada la *ISC*.
2. Ingresar los siguientes comandos para *detener el servidor*:
 - Para sistemas *Windows*, tipear: *la_raiz_isc \Runtime \AppServer \bin \stopServer.bat ISC_Portal* y presionar **Enter**.
 - Para sistemas *AIX, Linux, y Solaris*, tipear: *la_raiz_isc /Runtime /AppServer /bin /stopServer.sh ISP_Portal* y presionar **Enter**.

Para cualquier sistema *la_raiz_isc* es el directorio raíz para la instalación de la *ISC*.

3. Ejecutar el siguiente comando para *detener el sistema de ayuda*:

- Para sistemas *Windows*, tipear: `la_raiz_isc\Runtime\PortalServer\ISCEclipse\StopEclipse.bat` y presionar **Enter**.
- Para sistemas *AIX*, *Linux*, y *Solaris*, tipear: `la_raiz_isc/Runtime/PortalServer/ISCEclipse/StopEclipse.sh` y presionar **Enter**.

Para cualquier sistema, *la_raiz_isc* es el directorio raíz para la instalación de la ISC.

4.9 Instalar el Plug-in de la ISC para el Desarrollador de Aplicaciones

El programa de instalación de la *ISC* provee una opción para instalar el plug-in de la *ISC* para el *WebSphere Studio Application Developer V5.0*. El plug-in de la *ISC* incluye un asistente y un editor que hace fácil la creación de un archivo *component.xml* y agrega ese archivo a un proyecto de aplicación *portlet* existente.

Antes de instalar el plug-in, instalar el *WebSphere Studio Application Developer V 5.0*. Una vez que el *Application Developer* está instalado, el plug está incluido cuando se instala la *Integrated Solutions Console Toolkit*. Para usar el plug-in se necesita además el *WebSphere Portal Toolkit*, versión 4.2.5 o 4.3.

El plug-in provee documentación de ayuda que describe cómo usar el asistente y el editor, y cómo ejecutar manualmente algunas tareas que el plug-in no hace automáticamente. Para ver la documentación, clicar **Help** → **Help Contents** en la barra de herramientas para *WebSphere Studio Application Developer*. La ayuda se muestra en una ventana separada. Clicar la documentación de *Integrated Solutions Console* a la izquierda del esquema de navegación. Luego clicar el subtópico que se desea ver.

4.10 Verificación de la Instalación

Para verificar que la ISC se ha instalado satisfactoriamente:

1. Usar el navegador para abrir la URL de la *ISC*: `http://nombre.del.servidor:isc_port/ibm/console`, donde *nombre.del.servidor* es el nom-

bre del *host* para la instalación de la *ISC* y *isc_port* es el *puerto* para la ubicación de la *ISC* durante la instalación. Especificar el nombre del protocolo (*http*) en la URL, porque la URL contiene un número de puerto. Este número de puerto se encuentra en *la_raiz_isc \Runtime \isc.properties*. Dónde *la_raiz_isc* es el directorio raíz para la instalación de la *ISC*.

2. Iniciar sesión como administrador de la *ISC*. Especificar la ID del usuario configurado durante la instalación del Toolkit. La ID del usuario por defecto es *iscadmin*.
3. Clickear la opción **Settings**. Se muestra el esquema de navegación de los escenarios.
4. Clickear **User and Group Management**. La página se muestra en el área de trabajo.
5. Para ver la ayuda para los portlet en la página, clickear el ícono de ayuda portlet (el símbolo *?*). El tópico de ayuda de usuarios *Manage* muestra grupos en una ventana separada. Cerrar la ventana.
6. En la barra de herramientas de la *ISC*, clickear **Help**. Se inicia una ventana separada y se muestra un marco (frame) de navegación para acceder a la *Integrated Solutions Console Basics* de la *ISC* y al *Integrated Solutions Console Developer InfoCenter*.
7. Para cerrar sesión de la consola, click **Log off** en la barra de herramientas. Se muestra la página de Login.

4.11 Resolver Problemas de Instalación

Proceder con el siguiente chequeo si ocurren problemas durante la instalación de la *ISC*:

1. Verificar que se completaron todas las instrucciones de instalación.
2. Asegurarse de que están instalados los prerequisites correctos.
3. Asegurarse de que son correctas todas las configuraciones (tales como ID de usuario y contraseñas).

4. Chequear los mensajes de error en los archivos log de instalación.
5. Si el problema no ha sido resuelto, desinstalar la *ISC* y entonces instalarla nuevamente.

Bibliografía

- [1] L. Joyanes Aguilar. *Cibersociedad*. Mac Graw-Hill, 1997.
- [2] J. Kephart; D. Chess. *The Vision of Autonomic Computing*. IBM Corporation, Estados Unidos, 2003.
- [3] IBM Autonomic Computing. *Autonomic problem determination: A first step toward self-healing computing systems*. IBM Corporation, Estados Unidos, 2003.
- [4] IBM Autonomic Computing. *Autonomic Computing Toolkit User's Guide*. IBM Corporation, Estados Unidos, 2004.
- [5] N.Ñegroponte. *El Mundo Digital*. Ediciones B, Barcelona-España, 1995.
- [6] L. Small. *ABCs of the Autonomic Computing Toolkit*. IBM Corporation, Estados Unidos, 2004.
- [7] D. F. Bantz; C. Bisdikian; D. Challener; J. P. Karidis; S. Mastrianni; A. Mohindra; D. G. Shea; M. Vanover. *Autonomic Personal Computing*. IBM Corporation, Estados Unidos, 2003.

Índice de Materias

- AC
 - arquitectura, 15, 20
 - arquitectura de referencia, 16
 - estándares, 7
 - niveles
 - maduración, 20
 - niveles de maduración, 15
 - adaptable, 16
 - administrado, 15
 - autonómico, 16
 - básico, 15
 - predictivo, 16
 - paradigma, 7
- AC Information, 42
- AC Toolkit
 - Autonomic Computing Toolkit, 27
 - conceptos generales, 27
 - descarga, 48
 - desinstalación, 52
 - escenarios, 28, 39
 - herramientas, 28
 - información y documentación, 29
 - instalación, 48, 51
 - instalación de soluciones y tecnologías de despliegue, 35
 - paquete, 42
 - plataformas soportadas, 50
 - tecnologías, 28
 - tecnologías y herramientas, 31
 - tecnologías y soluciones, 29
 - uso, 30
 - Uso de, 29
- accesibilidad
 - del paradigma AC, 7
- Adapter Rule Editor
 - Editor de Reglas del Adaptador, 36
- administrador
 - autonómico, 17, 18
 - análisis, 18
 - ejecución, 18
 - monitoreo, 18
 - plan, 18
- administrador autonómico
 - virtual, 22
- AE
 - elemento autonómico, 21
- AIX, 53, 63
- AM
 - administrador autonómico, 21
- AME, 31, 33, 43
 - Autonomic Management Engine, 28
- APC
 - autonomic personal computing, 11
- autogestión
 - tecnología de, 5
- autonomic computing, 1, 2
- Autonomic Computing Toolkit, 27

- beneficios de la CA
 - a corto plazo, 8
 - a largo plazo, 9
- CA
 - computación autonómica, 2
- computación autonómica, 2
 - nociones, 1
 - personal, 20
- dependency checker
 - verificador de pre-requisitos, 31
- DNS, 44
- effector, 19
- escenario
 - de determinación de problemas, 39
 - de instalación automática, 41
 - SID, 47
- estabilidad, 25
- flexibilidad
 - del paradigma AC, 7
- Generic Log Adapter, 28
- GLA, 45
 - Generic Log Adapter, 35
- GLA/LTA, 45
- grid computing, 9
- Horn
 - Paul, 2
- IBM, 2, 6
- InstallAnywhere, 41
- Integrated Solutions Console, 28
 - Consola de Soluciones Integradas, 36
- interfase
 - de manejabilidad, 19
- Internet, 3
- IPSec
 - Internet Protocol Security, 24
- ISC, 44
 - arrancar y detener, 63
 - desinstalación, 63
 - instalación, 52, 54
 - instalación de plug in, 67
 - requerimientos para la instalación, 53
 - resolución de problemas de instalación, 68
 - secuencia gráfica de arranque y detención, 64
 - verificación de la instalación, 67
- ISMP, 41, 47
- IT, 4
 - information technologies
 - tecnologías de la información, 1
- Java, 6
- JVM, 54
- Linux, 53, 63
- log and trace
 - registro y seguimiento, 34
- Log and Trace Analyzer Tool, 28
- LTA, 45
 - Log and Trace Analyzer, 34
- Macrovision, 8
- middleware, 4
- multiplataforma, 5
- OASIS
 - Organization for the Advancement of Structured Information Standards, 7
- on demand
 - bajo demanda
 - a pedido, 5

- PDS
 - PDSscenario, 45
- privacidad, 24
- Problem Determination Scenario
 - Escenario de Determinación de Problema, 37
- recurso
 - administrado, 17, 19
- redes
 - auto-diagnosticadas, 2
 - auto-gestionadas, 2
 - transparentes, 2
- RMB, 34, 43
- SA
 - sistemas autonómicos, 4
- SAC
 - sistema de computación autonómica, 9
- SAS, 8
- SDD
 - Solution Deployment Descriptor, 7
- seguridad, 24
- sensor, 19
- servicio Web, 22
- SID, 44
- SIDS
 - Solution Installation and Deployment Scenario, 47
- SMD
 - editor, 35
 - Solucion Module Descriptor
 - Descriptor de Módulo de Solución, 35
- Solaris, 63
- SSL
 - Secure Socket Layer, 24
- TCPA
 - Trusted Computing Plataform Alliance, 24
- Tivoli, 24
- Toolkit, 7
- touchpoint, 19
- transparencia
 - del paradigma AC, 7
- UDDI
 - Universal Description, Discovery, and Integration, 22
- UI
 - unidad instalable, 41
- Whitehead
 - Alfred North, 1
- Windows, 53, 63, 64, 66, 67
- WMI
 - Windows Management Instrumentation, 21
- WSDM
 - Web Services Distributed Management, 8

