

Universidad de las Ciencias Informáticas

Facultad 4



**Título: Herramienta de instalación para el
Sistema de Gestión Penitenciaria Venezolano**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Joao Lemus Martínez

Yasser León Montes de Oca

Tutor: Ing. Jorge Ernesto Martínez Cabrera

Consultante: Ing. Madelín Haro Pérez

CIUDAD DE LA HABANA, MAYO DE 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2008.

Joao Lemus Martínez

Yasser León Montes de Oca

Ing. Jorge Ernesto Martínez Cabrera

AGRADECIMIENTOS

Agradecemos a todos los que han depositado su granito de arena en el presente trabajo. Gracias a todos por su preocupación, en especial a aquellas personas que lo revisaron y nos dieron su opinión. A Yoenia a Yanet a nuestro tutor, a Rosita, a Madelín, a nuestros compañeros de estudio, un millón de gracias por toda esa dedicación.

Joao y Yasser

DEDICATORIA

A mi familia, en especial mis padres, mis abuelos, mis tíos y tías, mis hermanos. A mis amigos. A Silvia Cristina que siempre estará en el corazón de todos nosotros...

Joao Lemus Martínez

A mi madre, por todo ese infinito cariño que me ha dado siempre, por ser: padre, madre, amiga y otras tantas cosas que llenan mi razón de ser. A mi hermana linda que siempre está cuidándome y que quiero con la vida. A mi novia (mi nene) que con su sonrisa me da cada día más. A mi tío papito, por quererme como a un hijo y por todo ese apoyo incondicional. A ellos, y a todos los que han depositado su confianza en mí, les dedico el presente trabajo.

Yasser León Montes de Oca

RESUMEN

La Universidad de las Ciencias Informáticas (UCI) ha venido desarrollando el Sistema de Gestión Penitenciaria (SIGEP) para la República Bolivariana de Venezuela. Existen aproximadamente 30 establecimientos penitenciarios distribuidos por los estados de este país. El sistema ha sido creado con el objetivo de contribuir a la mejora de las condiciones de vida de las personas que residen en estos establecimientos. En la actualidad la instalación de la aplicación se realiza de forma manual y poco intuitiva para los responsables de dicha actividad en cada centro. Por ello surge la necesidad de una herramienta de instalación que automatice las principales tareas de este proceso. El presente trabajo tiene como objetivo el desarrollo del instalador de *software* para el SIGEP.

Partiendo del proceso de despliegue y de la instalación de un *software* como su actividad fundamental, se ha hecho un estudio de los instaladores de aplicaciones así como las principales herramientas que brindan soporte para su creación. El modelo general de despliegue estudiado se ha particularizado en un conjunto de procesos para el caso específico del SIGEP. Para dar solución a la problemática existente se ha construido una herramienta de instalación utilizando *IzPack* como base para el desarrollo y *Apache Ant* como tecnología de automatización de código.

PALABRAS CLAVE

Despliegue, liberación, instalación, SIGEP.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción	4
1.2 El Proceso de Despliegue del Software.....	4
1.2.1 El Proceso de Despliegue del <i>Software</i> en RUP	5
1.2.2 El Proceso de Despliegue según IEEE	6
1.2.3 Actividades que se llevan a cabo en el Proceso de Despliegue.....	7
1.3 Instaladores de <i>Software</i>	9
1.3.1 Tipología.....	9
1.3.2 Herramientas para la creación de instaladores de <i>software</i>	10
Herramientas para <i>Windows</i>	10
Herramientas Open Source	12
1.3.3 Selección de la herramienta para la creación del instalador del SIGEP	14
1.4 Herramientas para la generación o automatización de código	15
1.5 Estado actual de la instalación de aplicaciones Web	16
1.6 ¿Cómo se instalan los productos desarrollados por la UCI con características similares al SIGEP?	17
1.7 Conclusiones	18
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	19
2.1 Introducción.....	19
2.2 Arquitectura de despliegue del SIGEP	19
2.3 Arquitectura de <i>software</i> del SIGEP	21
2.4 Principales problemas de la instalación manual del SIGEP.....	22

2.5	Características que debe tener la instalación del <i>software</i> SIGEP	23
	Precondiciones de la instalación.....	23
	Parámetros configurables	23
2.6	Flujo de procesos de la instalación del SIGEP	24
	Entidades comunes que se manejan en los procesos de la instalación del SIGEP.	25
2.6.1	Liberación de la aplicación Web	26
	Principales entidades que intervienen	26
	Tareas que conforman la liberación Web	27
2.6.2	Inserción del sistema en el lado del cliente	29
2.6.3	Ejecución de componentes	32
2.6.4	Desinstalación de la aplicación	33
2.6.5	Actualización de una versión del sistema	34
2.7	Conclusiones	34
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....		35
3.2	Construcción de la liberación Web	35
3.2.1	Configuración del archivo de construcción	35
3.3	IzPack como herramienta de construcción	36
3.3.1	Arquitectura de <i>IzPack</i>	36
3.3.2	Configuración del archivo de instalación	38
	Información general	39
	Especificaciones de los componentes GUI.....	39
	Internacionalización	40
	Paneles.....	41
	Recursos.....	42
	Paquetes.....	43
3.4	Integración de componentes.....	44
3.5	Simulación del entorno real. Resultado obtenido.....	45
3.5.1	Resultado obtenido.....	45
3.6	Conclusiones	46

CONCLUSIONES	47
RECOMENDACIONES	48
BIBLIOGRAFÍA	49
ANEXOS	50
Anexo # 1 Ventanas de información y configuración de la herramienta de instalación del SIGEP	50
GLOSARIO	53

TABLA DE FIGURAS

Figura 1- El Despliegue como flujo de trabajo en RUP	5
Figura 2 - Actividades del proceso de despliegue de <i>software</i>	7
Figura 3 - Comparación entre las herramientas para la creación de instaladores	14
Figura 4 - Arquitectura de distribución del SIGEP	19
Figura 5 - Distribución de las capas de <i>software</i>	21
Figura 6 - Flujo de procesos de la instalación del SIGEP	25
Figura 7 - Modelo lógico de la liberación Web	26
Figura 8 - Flujo de tareas para la liberación de la aplicación Web del SIGEP	28
Figura 9- Flujo de tareas para la inserción del SIGEP en el lado del cliente	29
Figura 10 - Flujo de instalación de la aplicación	31
Figura 11 - Flujo de tareas de desinstalación de la aplicación.....	33
Figura 12 - Arquitectura de compilación de <i>IzPack</i>	37
Figura 13 - Estructura de componentes de <i>IzPack</i>	38

INTRODUCCIÓN

En la República Bolivariana de Venezuela se llevan a cabo una serie de transformaciones políticas, económicas y sociales para erradicar fenómenos como la pobreza, el analfabetismo, la escasez de servicios médicos y la violencia, entre otros males sociales. Dentro de estas transformaciones, el gobierno venezolano ha formulado un proyecto de Humanización del Sistema Penitenciario por ser unos de los sectores más afectados y para mejorar la crítica situación en la que están sumergidos los establecimientos penitenciarios venezolanos. Como parte de este proyecto, la Universidad de las Ciencias Informáticas (UCI), desde septiembre de 2006 ha venido desarrollando el SIGEP, *software* que apoyaría la actividad de los Establecimientos Penitenciarios, Unidades Técnicas de Apoyo al Sistema Penitenciario (UTASP) y Centros de Tratamientos Comunitarios (CTC), ayudando a resolver las necesidades de la gestión de información y la toma de decisiones por parte de la Dirección General de Servicios Penitenciarios (DGSP).

En la actualidad existen aproximadamente 30 establecimientos penitenciarios (Internados Judiciales, Centros Penitenciarios, Centros de Tratamiento Comunitario) y otras sedes (Coordinaciones Regionales y las UTASP) distribuidos por los distintos estados de la hermana República Bolivariana. Dentro del proceso de despliegue en cada uno de estos establecimientos, la instalación del SIGEP se realiza a partir de un componente que contiene todo el código de la aplicación y cada una de las configuraciones específicas para el centro en el que se está instalando el *software*. Esto impone que para cada centro penitenciario se cree un componente distinto con sus configuraciones particulares, entiéndase esto como: dirección de la base de datos, dirección física donde se guardarán las trazas de auditoría y el establecimiento en el que se instalará el sistema.

El SIGEP está constituido por varios subsistemas que representan cada una de las áreas del sistema penitenciario venezolano. Estos subsistemas a la vez se componen de módulos, los cuales contienen las funcionalidades que responden a los procesos de dichas áreas. Existen dos casos en los cuales la aplicación no se requiere en su totalidad: no se tiene una versión estable del producto o no se necesitan algunos de estos subsistemas y módulos en el penal en que se instala el sistema. Para ambos, se tienen que deshabilitar los módulos que no se desean incluir, modificando manualmente los ficheros de configuración. Actualmente se despliegan estos módulos sin necesidad alguna, lo cual trae consigo el desaprovechamiento de los recursos en el proceso de despliegue tales como la memoria física del servidor, el tiempo de instalación y la existencia física de subsistemas y módulos innecesarios para el funcionamiento de la aplicación en el servidor de un determinado centro penitenciario.

Hasta la fecha actual la instalación del sistema se concibe de forma desacoplada. Existe una separación entre los componentes utilizados en dicha actividad tales como: la máquina virtual de Java (JVM¹), el servidor de aplicación Web *Apache TomCat* y el fichero que contiene la aplicación. Se verifican manualmente las precondiciones fundamentales para instalar el *software* tales como: el sistema operativo, la existencia de una plataforma y un servidor Web, entre otras. El personal encargado de llevar a cabo este procedimiento necesita tener vastos conocimientos sobre tecnologías J2EE², servidores de aplicaciones Web, servidores de bases de datos así como características puntuales de la arquitectura del sistema para poder hacer las configuraciones pertinentes.

La instalación del SIGEP se hace complicada y riesgosa; es necesario llevar a cabo una serie de configuraciones, que al hacerlas manualmente, se puede incurrir en errores como la equivocación al habilitar o deshabilitar los módulos del sistema, la entrada incorrecta de las configuraciones de la base de datos y del identificador del centro penitenciario. Todo esto puede conllevar a iniciar de nuevo el proceso de instalación y configuración. También se debe tener en cuenta que la instalación consume mucho tiempo por lo que se prolonga el proceso de despliegue. Estos problemas e inversión de tiempo se podrían erradicar y minimizar considerablemente si se automatizara la instalación y configuración del sistema.

Tomando en cuenta la situación anterior se plantea el siguiente problema:

¿Cómo automatizar las tareas referentes a la instalación de *software* en el proyecto SIGEP de manera que cumpla con las exigencias de despliegue tales como continuidad y parametrización?

El objeto de estudio y el campo de acción lo constituyen el proceso de despliegue de las aplicaciones Web y la instalación del SIGEP respectivamente.

Para dar solución al problema anteriormente planteado se ha trazado como objetivo general la creación de la herramienta de instalación que apoye el despliegue del SIGEP.

Para cumplir el objetivo y solucionar la problemática de esta investigación se trazaron las siguientes tareas:

- Estudiar la evolución y desarrollo de los instaladores de *software*.
- Investigar el funcionamiento y las características de las principales herramientas de desarrollo de instaladores.

¹ JVM, del inglés *Java Virtual Machine*: Intérprete para el lenguaje de máquina de Java.

² J2EE, del inglés *Java 2 Platform, Enterprise Edition*: Tecnología para desarrollar aplicaciones empresariales basadas en la Web.

- Establecer una comparación entre las diferentes herramientas investigadas para su posterior aplicación.
- Identificar los requisitos que debe cumplir el proceso de instalación del SIGEP.
- Modelar el flujo de tareas para lograr los objetivos planteados.
- Implementar la herramienta de instalación del SIGEP de acuerdo con el flujo de tareas modelado.

Este trabajo está conformado por tres capítulos que recogen todo lo abordado en la investigación:

- Capítulo 1: Fundamentación Teórica, se brinda un estado del arte sobre los elementos que forman parte del problema que se presenta, así como aquellos que son importantes a tener en cuenta para dar la solución que se requiere.
- Capítulo 2: Características del Sistema, describe la situación problemática y la propuesta de solución. Además se especifican las características que debe tener el mecanismo de instalación, se propone un flujo de tareas y se identifican las entidades que interactúan en dichas tareas.
- Capítulo 3: Construcción de la Solución Propuesta, aborda los detalles relacionados con la construcción del instalador y de cada uno de los elementos que lo conforman.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se presentan las definiciones y conceptos fundamentales relacionados a los procesos que enmarcan el problema a resolver de la investigación en curso. Se presenta un estado del arte de cada una de las herramientas que brindan un soporte para la creación de instaladores de *software*. Además se establece una comparación entre los exponentes fundamentales de estas herramientas. Se describen las principales tecnologías para la generación o automatización de código así como el estado actual de la instalación de las aplicaciones Web.

1.2 El Proceso de Despliegue del Software

Las aplicaciones de *software* actualmente no son sistemas autónomos. Son cada vez más el resultado de la integración de colecciones de componentes heterogéneos, muchas veces distribuidos sobre una red informática. Estos componentes pueden ser proporcionados por diferentes proveedores y pueden ser parte de diversos sistemas al mismo tiempo. Los componentes cambian y se transforman muy rápidamente, lo que hace difícil gestionar todo el sistema de manera coherente. En este escenario, un paso crucial del ciclo de vida del *software* es el proceso de despliegue. Informalmente, el término de despliegue se refiere a todas las actividades que hacen que un sistema esté disponible para sus usuarios. Si bien esto es razonable e intuitivo, la creación de un marco de evaluación para el despliegue de tecnologías, requiere una precisa y amplia comprensión de la naturaleza y las características del despliegue de *software* como proceso.

El despliegue de un sistema de *software* involucra la transferencia o copia de sus componentes desde el lado del proveedor hacia el lado del cliente, este constituye su principal objetivo. Una vez desplegado, dicho sistema estará disponible para su uso en el lado del cliente.

El proceso de despliegue de *software* puede ser definido como la entrega, el ensamblaje y la gestión, en un determinado sitio, de los recursos necesarios para utilizar una versión de un sistema [1].

1.2.1 El Proceso de Despliegue del Software en RUP

Este proceso está contenido dentro de los nueve flujos de trabajo básicos en el Proceso Unificado de Software (RUP) formando parte de las fases de Elaboración, Construcción y Transición de esta metodología de desarrollo. En la figura # 1 queda resaltado dicho comportamiento.

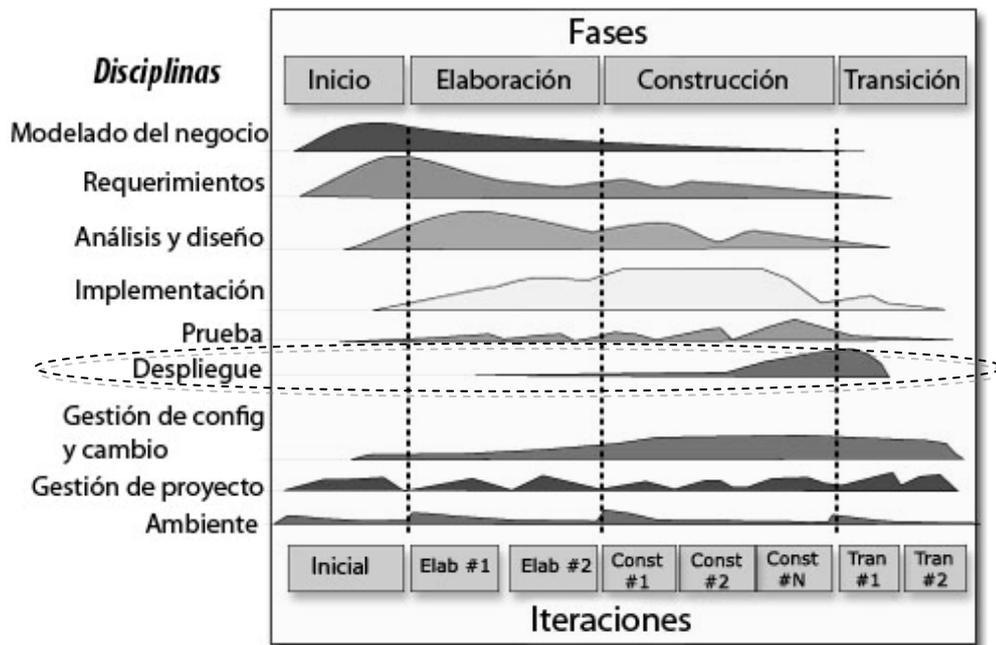


Figura 1- El Despliegue como flujo de trabajo en RUP

El objetivo de este flujo de trabajo para RUP es producir con éxito versiones de los productos y entregar el *software* a sus usuarios finales. Este abarca una amplia gama de actividades que incluyen:

- Producción de versiones externas del *software*.
- Empaquetado del *software*.
- Distribución del *software*.
- Instalación del *software*.
- Prestación de ayuda y asistencia a los usuarios.

En muchos casos, también se incluyen actividades tales como:

- Planificación y realización de pruebas beta.
- Migración de los actuales programas o datos.
- Aceptación formal.

Aunque las actividades de despliegue son en su mayoría centradas en la fase de transición, muchas de ellas necesitan ser incluidas en fases anteriores para preparar el despliegue a finales de la fase de construcción.

1.2.2 El Proceso de Despliegue según IEEE³

Para el estándar de desarrollo de procesos de *software* IEEE, el despliegue, al cual se refieren con el término de grupo de actividades posteriores al desarrollo [4], tiene definido como sus principales actividades la instalación, la operación y soporte, el mantenimiento y la retirada de un determinado producto de *software*.

Las actividades de instalación consisten en la transportación e instalación de un sistema de *software* desde el entorno de desarrollo hasta el cliente. Incluyen la correcta distribución del producto, las modificaciones necesarias, el chequeo y la aceptación en un ambiente operacional. Si surge un problema, deberá ser identificado y reportado [4].

Por otra parte, la operación y el soporte implican la utilización del sistema por parte del usuario y el apoyo permanente a este. El soporte incluye la prestación de asistencia técnica, consultoría con el usuario, y el registro de las solicitudes de los usuarios [4].

El grupo de actividades de mantenimiento se refiere a la identificación de las mejoras y la resolución de los errores de *software*, fallos y fracasos. Las principales tareas son: la identificación de las necesidades de mejora del producto y la aceptación de cualquier anomalía en un recurso, reportándola como un posible problema [4].

Por último, la retirada de una aplicación informática supone la eliminación de un sistema existente, ya sea por el cese de su funcionamiento o de apoyo, o por su sustitución por un nuevo sistema o una versión actualizada del sistema existente [4].

De una forma más sintetizada se puede decir que el despliegue de *software* son todas las actividades posteriores al desarrollo que se llevan a cabo para producir versiones exitosas de un determinado sistema, teniendo en cuenta también el mantenimiento de este y la migración de tecnología si fuera necesaria.

³IEEE, del inglés *Institute of Electrical and Electronics Engineers*: asociación técnico-profesional mundial dedicada a la estandarización en las ciencias de la computación, telecomunicaciones, tecnologías biomédicas y otras ramas de las ciencias.

1.2.3 Actividades que se llevan a cabo en el Proceso de Despliegue

El proceso de despliegue es generalmente descrito como un ciclo de vida que involucra varios subprocesos o tareas interrelacionadas posteriores al desarrollo, tales como: liberación, instalación, activación, desinstalación, desactivación y actualización [2] (figura # 2). La creciente complejidad de los sistemas de *software* requiere que a estas actividades del despliegue se les de más atención.

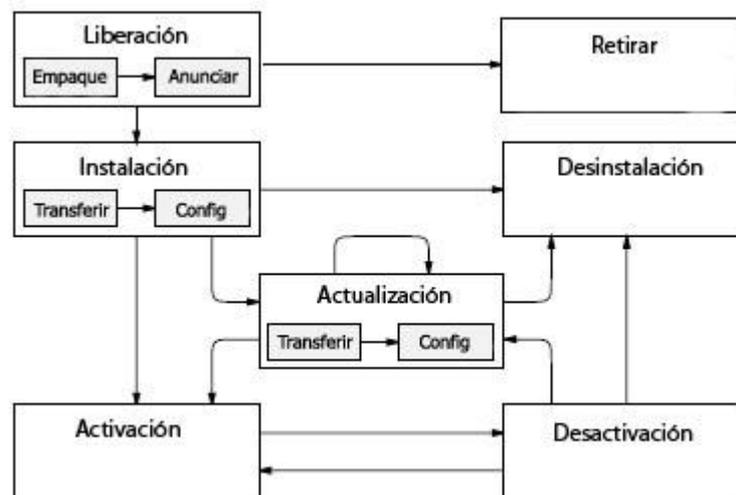


Figura 2 - Actividades del proceso de despliegue de *software*

Aunque se puede identificar un conjunto de las principales actividades que constituyen un proceso de despliegue genérico, no se puede precisar las prácticas particulares y los procedimientos específicos de cada una. Se dependerá en gran parte de la naturaleza del *software* a ser liberado, y de las características y requisitos de los proveedores y clientes. Por lo tanto, el proceso descrito en la figura # 2 se debe interpretar de una manera razonable, debe que ser personalizado y enriquecido de acuerdo a requisitos específicos de despliegue de la actividad que se observó.

A continuación se presenta una descripción de cada una de las actividades del proceso de despliegue:

Liberación: Constituye la interfaz entre el proceso de desarrollo y el proceso de despliegue. Abarca todas las operaciones necesarias para preparar un producto y que pueda ser correctamente implantado en el sitio del consumo. Así, la liberación debe determinar todos los recursos necesarios por un sistema de *software* para operar correctamente en el sitio de consumo. También debe recoger toda la información necesaria para la realización de actividades posteriores del proceso de despliegue. Esta información puede ser derivada de una variedad de fuentes, entre ellas el proceso de desarrollo y el conocimiento humano sobre la estructura y funcionamiento del sistema. La liberación incluye el

empaquetado del producto para que pueda ser transportado. Este paquete debe contener los componentes y una descripción del sistema que incluye los requisitos y las dependencias de otros componentes externos, los procedimientos de instalación, y toda la información que sea relevante para la gestión de la aplicación. Otro paso en la liberación es la capacitación, es decir, el conjunto de operaciones que se necesitan para difundir la información pertinente a las partes interesadas en las características y la utilización del sistema que se va a desplegar [3].

Instalación: Comprende la inserción inicial de un sistema en el lado del cliente. Por lo general, es la más compleja de las actividades de despliegue, ya que se ocupa de la correcta concentración de todos los recursos necesarios para utilizar el producto de *software*. Con el término Instalación se hace referencia a dos tareas principales. La primera de ellas es la transferencia y entrega del producto desde el proveedor hasta el cliente en el sitio a ser utilizado. La segunda consiste en todas las operaciones de configuración que son necesarias para que el sistema esté listo para la activación [3].

Activación: Se refiere a la ejecución de aquellos componentes de un sistema que son necesarios para el funcionamiento del producto que se va a desplegar. Para una herramienta simple, la activación involucra el establecimiento de alguna forma de mando (hacer *clic* en el icono gráfico) para la ejecución de componentes binarios de la herramienta. Para un sistema complejo, podría ser necesario iniciar servidores demonios⁴ antes de que el sistema de *software* sea activado. Tener en cuenta que el proceso de instalación en sí mismo, puede requerir la invocación de otros instrumentos y posiblemente, la instalación de estos [3].

Desactivación: Constituye la actividad inversa de la activación, y se refiere al cierre de cualquier componente en ejecución de un sistema instalado. En general, la desactivación es necesaria antes de que otras actividades de despliegue se puedan llevar a cabo tales como la actualización y la desinstalación [3].

Desinstalación: En algún momento, un sistema en su conjunto ya no es necesario en un determinado sitio de consumo y puede ser eliminado. Se supone que la desinstalación está precedida por la desactivación. La actividad posiblemente implique una modificación de las configuraciones de otros sistemas, así como la retirada de los archivos pertenecientes a la aplicación que ha de ser desinstalada. La desinstalación no es necesariamente un proceso trivial. Una de las razones es que este proceso podrá asumir ser el único usuario de algún recurso compartido, como por ejemplo

⁴demonio (en inglés *daemon*, *Disk And Execution Monitor*): tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

diversos archivos de datos, por lo que pueden liberar esos recursos y causar que otros sistemas puedan fallar [3].

Actualización: Esta actividad constituye un caso especial de la instalación. Es por lo general menos compleja debido a que muchos de los recursos necesarios ya se han obtenido durante el proceso de instalación. Normalmente, el ciclo de vida del despliegue incluye una secuencia repetida en el que un sistema es desactivado, se instala una nueva versión y este vuelve a ser activado. Para algunas aplicaciones, la desactivación puede no ser necesaria y la actualización se puede realizar al mismo tiempo donde una versión anterior esta aún activa. De manera similar a la instalación, la actualización incluye la transferencia de todos los componentes necesarios para completar la operación [3].

Todas estas actividades forman parte de un mismo proceso lógico. En el despliegue de un producto informático, la integración de cada una de las actividades anteriormente descritas es fundamental. En este punto se necesita de un mecanismo que acople todo este proceso lógico de una forma automatizada. La herramienta que responde a esta necesidad es conocida como instalador de *software*.

1.3 Instaladores de *Software*

En la actualidad existe un gran número de instaladores de *software*, es muy inusual que una aplicación se encuentre sin su respectivo instalador. Es necesario desplegar una aplicación sin incurrir en configuraciones demasiado engorrosas que no son de la incumbencia de los usuarios finales. Estos usuarios no están obligados a tener conocimientos avanzados para instalar un determinado producto.

1.3.1 Tipología

De manera informal un instalador es un programa de computación que instala archivos, tales como aplicaciones, controladores de dispositivos, u otro tipo de *software*, al ordenador [7]. De acuerdo a la gran variedad de aplicaciones en el mundo de la informática y a sus características fundamentales, en esta investigación los instaladores se clasifican en dos grandes grupos:

- Instaladores estándares.
- Sistemas de gestión de paquetes.

Los instaladores estándares están específicamente realizados para desplegar los archivos que contienen. Algunos de estos son de uso general y su función consiste en leer el contenido del paquete de *software* que va a ser instalado.

Un *paquete de software* es un conjunto de uno o más archivos que son necesarios para la ejecución de un programa de computación o añadir a las características de un programa ya instalado en uno o más ordenadores [8].

Linux y otros sistemas Unix suelen administrar miles de paquetes. Los sistemas de gestión de paquetes, también conocidos como gestor de paquetes o sistema gestor de instalación, son una colección de herramientas para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de un ordenador. Los paquetes son las distribuciones de *software* y los metadatos, como el nombre completo, la descripción de su objeto, el número de versión, los proveedores, y una lista de las dependencias necesarias para que el sistema pueda funcionar correctamente. Tras la instalación, los metadatos se almacenan en una base de datos de paquetes locales. Un sistema de gestión de paquetes proporciona un método coherente para la instalación de una aplicación informática.

Los instaladores que predominan son para aplicaciones de escritorio. Estos son mucho más fáciles de desarrollar debido a que existe toda una gama de herramientas para su confección. En el caso particular de las aplicaciones Web, esta cantidad se ve mucho más restringida. Normalmente no existe una herramienta que automatice el proceso de instalación por completo. En algunos casos, si existe dicha herramienta, esta requiere de configuraciones manuales que hacen el proceso muy complicado. La mayoría de las veces los procesos de instalación para las aplicaciones Web se llevan a cabo por un personal especializado en las características propias de la aplicación.

1.3.2 Herramientas para la creación de instaladores de *software*

En la actualidad existe toda una infinidad de instaladores, cada uno de estos se ha construido sobre la base de tecnologías y herramientas que están en constante desarrollo. Se pueden encontrar bajo licencias comerciales, *software* propietario, o expuestas a las distintas comunidades de desarrollo como código abierto, libre o gratuito.

Herramientas para *Windows*

Un popular formato de *Microsoft Windows* es el paquete de instalación MSI⁵, el cual es instalado por la herramienta de empaquetamiento de *software Windows Installer*. Diferentes empresas comerciales han desarrollado herramientas con el propósito de crear instaladores de aplicaciones para el sistema

⁵MSI, del inglés *Microsoft Installer*: los paquetes MSI sirven para poder instalar aplicaciones directamente en el equipo sin pedir interacción alguna por parte del usuario.

operativo *Windows* tales como *InstallShield*, *InstallAnywhere*, *Wise*, *SetupBuilder* y *Desktop Authority MSI Studio*, entre otras. La mayoría de estas herramientas pueden crear paquetes MSI, así como sus propios ejecutables.

Microsoft Windows Installer Service (WIS). Es un poderoso servicio que ha sido creado por *Microsoft* para ayudar a la gestión del ciclo de vida de una aplicación en el sistema operativo *Windows*. Hoy, *Windows Installer* está en su cuarta edición y brinda un consistente punto de interacción para la instalación de aplicaciones comerciales y de auto consumo. Un aspecto importante de *Windows Installer* es que brinda una serie de funcionalidades que han estado deshabilitadas en sistemas de instalación convencionales.

InstallShield es el estándar de la industria para instalaciones con *Windows Installers* e *InstallScript*. Permite a los productores de *software* realizar instalaciones de alta calidad para las plataformas *Windows*, incluyendo *Vista™*, y ampliarlas para configurar servidores de bases de datos, servicios Web y dispositivos móviles. *InstallShield* está disponible actualmente en tres ediciones comerciales: de primera, profesional y *Express*. También está disponible en tres idiomas: inglés, alemán y japonés.

SetupBuilder™ ha sido diseñado para ofrecer una fusión de la instalación, el despliegue Web, la gestión de la configuración y las tecnologías de *script*. Su versión denominada *Developer Edition* proporciona características avanzadas para la instalación y actualización Web además de un depurador visual para la resolución de errores mediante el control de sus instalaciones, soporte a múltiples lenguajes, entre otras. Su versión profesional es una herramienta indispensable para los desarrolladores que necesiten una suite de instalación más sofisticada con capacidad para la escritura de *scripts*. Incluye una funcionalidad para la creación de instaladores de parches con el objetivo de actualizar las versiones antiguas de las aplicaciones y muchas otras funciones avanzadas, incluyendo soporte a DLL⁶ y el llamado a las API⁷ de *Windows*, además de un editor de ficheros *script*. *SetupBuilder™* es un rápido desarrollador de instaladores y una herramienta de gestión de la configuración para sistemas operativos *Windows*, tales como: *Windows Server 2008 ("Longhorn")*, *Windows Vista*, *Windows 2003*, *Windows XP*, *Windows 2000*, *Windows NT4* y *Windows Me/98/95*. Ofrece el conjunto completo de las características que se necesitan para construir instaladores y actualizadores robustos para este sistema operativo. Se trata de una potente, flexible, rápida e intuitiva herramienta basada en *scripts* utilizada en la gestión de configuración.

⁶DLL, del inglés *Dynamic Linking Library*: término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.

⁷API, del inglés *Application Programming Interface*: es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

Desktop Authority MSI Studio™ es una aplicación de empaquetado e instalación de *software* y una solución optimizada en la gestión de instaladores para el uso de los administradores de sistemas. Cuenta con un conjunto completo de características para capturar, crear y editar archivos MSI. Presenta un mecanismo que controla el comportamiento de los paquetes de instalación en *Windows*. Además, la edición profesional permite probar y validar los paquetes sin que ello afecte a los sistemas de producción.

InstallAnywhere® posibilita a los desarrolladores de *software* de una manera rápida y fácil, instalaciones para *UNIX*, *Linux*, *Solaris*, *HP-UX*, *Mac OSX*, *Windows* entre otros sistemas operativos. Ahorra tiempo y reduce los costos mediante la creación de un único archivo de proyecto que genere configuraciones fiables para cada una de las plataformas mencionadas. *InstallAnywhere®* se encuentra disponible en dos versiones (*Standard* y *Enterprise*) y en cuatro idiomas (inglés, alemán, francés y japonés).

Herramientas Open Source⁸

Alternativas libres incluyen *NSIS*, *Clickteam Install Creator* y *InnoSetup*, así como una herramienta de *Microsoft* llamado *WiX*. *BitRock* tiene una herramienta multiplataforma, llamado *InstallBuilder*, que crea instaladores para *Windows*, *Mac OSX*, *Linux* entre otros; y es gratuita para los proyectos de código abierto. La comunidad de desarrolladores de *Java* también se interesan en este aspecto y ha creado *IzPack*, herramienta que también permite generar instaladores multiplataforma.

Dentro de las alternativas libres mencionadas *NSIS (Nullsoft Install scripts System)* es un sistema profesional de código abierto para crear instaladores sobre *Windows*. Está diseñado para construir instaladores pequeños y flexibles tanto como sea posible y, por lo tanto, es muy apropiado para la distribución por Internet. *NSIS* soporta diferentes idiomas, está basado en *scripts* y permite generar la lógica para manejar incluso las más complejas tareas de instalación. Muchos *plugins*⁹ y *scripts* ya están disponibles, con los que se pueden crear instaladores Web, comunicarse con *Windows* y otros componentes de *software*, instalar o actualizar elementos compartidos.

Clickteam Install Creator permite crear sistemas de instalación llevando todo el proceso paso a paso a través de un asistente. Estos pasos comprenden la selección de ficheros que incluye el programa, la elección del directorio donde se instalarán, los posibles acuerdos de licencia, hasta llegar al último paso, que compila la instalación y la deja lista para usar. *Clickteam Install Creator* permite una

⁸Open source: término en inglés con el que se conoce al *software* distribuido y desarrollado libremente.

⁹Plugin: aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

instalación personalizada, desde los textos y logos que aparecen hasta la inclusión de opciones adicionales como registro de componentes (DLL, OCX, REG y TLB) o creación de iconos de acceso directo a la aplicación.

Inno Setup es otro instalador gratuito para programas de *Windows*. Presentado por primera vez en 1997, hoy *Inno Setup* compite e incluso supera muchos instaladores comerciales en cuanto a características y estabilidad. Soporta todas las versiones de *Windows* desde Vista, XP hasta NT 4.0.

*Windows Installer XML*¹⁰ (WiX) es un conjunto de herramientas que construye los paquetes de instalación sobre *Windows* basado en ficheros XML. Este conjunto de herramientas soporta un entorno de comandos que los desarrolladores pueden integrar en sus procesos de construcción para crear paquetes de instalación MSI y MSM. Fue liberado por *Microsoft* bajo una licencia de código abierto denominada *Common Public License*.

BitRock es otra de las herramientas que permite crear instaladores multiplataforma que se pueden compilar para *Windows*, *Linux*, *Mac*, *FreeBSD*, *Solaris* (x86/Sparc), *IRIX*, *AIX* y *HP-UX*. Los instaladores generados tienen un nativo "*look-and-feel*" y no requiere de dependencias, puede funcionar en modo gráfico (GUI), en modo texto y en modo desatendido. Esta herramienta de instalación puede también generar paquetes RPM independientes.

IzPack es un creador de instaladores basado en tecnología Java. Es un proyecto *SourceForge* cuyo diseño modular permite definir cómo debe verse el instalador, además es muy configurable mediante un API. Dentro de sus características fundamentales se encuentra el soporte multiplataforma que le permite ejecutarse en cualquier sistema operativo que tenga instalada la máquina virtual de Java. Esta herramienta está bajo la licencia GPL (*GNU General Public License*), sus cláusulas dictaminan que puede modificarse, usarse y redistribuirse gratuitamente, o sea que se puede modificar el software y publicarse de forma gratuita pero con la versión modificada es obligatorio compartir también el código cambiado.

Mac OSX tiene una aplicación incluida llamada *Installer*. Sin embargo, esta no es técnicamente un instalador, sino un gestor de paquetes. Algunas aplicaciones comerciales para Mac OSX utilizan un instalador personalizado, generalmente *Installer VISE* o *Stuffit InstallerMaker*.

¹⁰XML, del inglés *Extensible Markup Language*: metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C).

1.3.3 Selección de la herramienta para la creación del instalador del SIGEP

Dadas las herramientas anteriormente descritas, a continuación se muestra una comparación de acuerdo a las principales características que requieren de nuestra atención. La plataforma de instalación, plataforma de desarrollo, el soporte a diversos lenguajes, licencia, la creación de una herramienta de desinstalación, tipo de archivos que se manejan, la presencia de componentes gráficos (GUI) son los principales criterios de comparación a medir entre los distintos tipos de herramientas.

Producto	Licencia	Plataformas de instalación		Plataformas de desarrollo		Multilenguaje	Desinstalación	Modos de instalación		Tipo de Archivos
		Linux	Windows	Linux	Windows			Texto	Gráfico	
InstallShield	Comercial	X	X	X	X	X	X	X	X	Script
InstallAnywhere	Comercial	X	X	X	X	X	X	X	X	Script
SetupBuilder	Comercial		X		X	X			X	Script
Desktop Authority MSI Studio	Comercial		X		X	X			X	Script
NSIS	CPL		X	X	X	X	X	X	X	Script
Clicteam Install Creator	Libre		X		X		X		X	Script
InnoSetup	Open source		X		X	X	X	X	X	Script
BitRock Install Builder	Open Source	X	X	X	X	X	X	X	X	XML
IzPack	GPL	X	X	X	X	X	X		X	XML
Windows Installer XML (WiX)	CPL		X		X		X		X	XML
CreateInstall	Comercial		X		X	X	X		X	Script
GkSetup	Comercial		X		X	X	X		X	Script

Figura 3 - Comparación entre las herramientas para la creación de instaladores

De acuerdo al contenido de la tabla en la figura # 3 se puede decir que la totalidad de las herramientas mostradas cumplen con la mayoría de los parámetros a comparar. La mayor diferencia radica en la licencia bajo la cual estos productos están publicados, el soporte para plataformas Linux, los modos de instalación y el tipo de ficheros que manejan dichas herramientas.

Es necesario resaltar que la selección de una de estas herramientas para dar solución al problema que ocupa la investigación en curso debe ser consecuente con las políticas a las cuales se adhiere la solución del SIGEP. Tanto el cliente de este producto de software, la Dirección General de Servicios Penitenciarios; como sus desarrolladores en la UCI, están interesados en una aplicación

completamente *open source*. Por lo anteriormente descrito se podrían descartar las herramientas cuyas licencias son comerciales y no son de interés para esta investigación. Por otra parte, el despliegue del SIGEP se hace sobre la plataforma Linux lo que deja solo dos propuestas dentro de las herramientas analizadas: *IzPack* y *BitRock*. Ambas herramientas tienen características similares como el soporte multiplataforma, soporte a múltiples lenguajes, el tipo de archivos que manejan (XML), la presencia de componentes GUI y la generación de una herramienta de desinstalación.

Tanto *IzPack* como *BitRock* soportan aplicaciones escritas en el lenguaje Java y la integración con herramientas de generación o automatización de código (ver epígrafe 1.5). *IzPack* es mucho más robusto en este sentido. Es un producto creado completamente sobre el lenguaje de programación Java, lo cual permite explotar sus potencialidades. Esto, unido a que el SIGEP está desarrollado sobre el mismo lenguaje y a las características particulares de esta solución de software, hace necesario que la herramienta para crear su instalación sea extensible para este lenguaje, lo cual deja a *IzPack* como herramienta a utilizar en el presente trabajo.

1.4 Herramientas para la generación o automatización de código

Muchas de las herramientas anteriormente mencionadas están diseñadas para la integración con tecnologías de generación o automatización de código, compactando así el proceso de desarrollo de cada uno de los componentes que conforman en su totalidad los instaladores de aplicaciones. Los principales exponentes de estas tecnologías son *Make*, *Apache Ant* y *Maven*.

Make es una herramienta de generación o automatización de código, muy usada en los sistemas operativos tipo *Unix/Linux*. Es una utilidad que permite definir reglas de dependencia entre ficheros. Aunque puede utilizarse para diferentes fines, está especialmente orientada a la compilación de código. El propósito de *Make* es determinar automáticamente las piezas de un programa que necesitan ser recompiladas y lanzar las órdenes necesarias para lograrlo. Por defecto lee las instrucciones para generar el programa u otra acción de un determinado fichero (*makefile*). Las instrucciones escritas en este fichero se denominan dependencias. La herramienta *Make* se usa para las labores de creación de un fichero ejecutable o programa, su instalación, la limpieza de los archivos temporales en la creación del fichero, entre otras funcionalidades, como la creación de documentos del formato *docbook*¹¹ y el mantenimiento del sistema. Todo esto lo realiza usando o creando los llamados *makefiles* que ejecuten cada una de estas tareas.

¹¹ docbook: dialecto de SGML (Standard Generalized Markup Language) que permite la escritura de documentación técnica.

Por otra parte, *Maven* es una herramienta para la gestión y compresión de proyectos Java. Estaba integrado dentro de *Jakarta* pero ahora es una tecnología de nivel superior de la *Apache Software Foundation*. Esta herramienta puede compilar un proyecto Java, ejecutar pruebas unitarias, generar paquetes (JAR, WAR, EARS o distribuciones en ZIP) y una serie de reportes. Una de sus características más importantes es su actualización en línea mediante servidores repositorios. *Maven* es capaz de descargar nuevas actualizaciones de las bibliotecas de las que depende el proyecto y de igual manera subir una nueva distribución a un repositorio de versiones, dejándola a disposición de todos los usuarios.

Otro ejemplo de este tipo de herramientas es *Apache Ant (Another Neat Tool)* la cual es una tecnología basada en Java. Se utilizan en el desarrollo de software para transformar el código fuente y otros archivos de entrada en un formato ejecutable. *Ant* es similar a *Make* en cómo define las dependencias entre tareas de construcción, sin embargo, en lugar de implementarlas por medio de una plataforma específica, utiliza la plataforma Java. Con *Apache Ant*, se puede escribir un único fichero de construcción que opera constantemente sobre cualquier plataforma Java, lo que constituye su mayor fortaleza. Otros puntos fuertes a tener en cuenta son su simplicidad y su poderosa capacidad de ser extendido.

Dada las características de estas tres herramientas para la generación o automatización de código, y teniendo en cuenta la comparación entre las distintas herramientas para la creación de instaladores de *software* anteriormente descrita se llega a la conclusión que la más aceptada para esta investigación es *Apache Ant*.

1.5 Estado actual de la instalación de aplicaciones Web

Originalmente las aplicaciones Web construidas por desarrolladores eran instaladas por los mismos desarrolladores o administradores de sistemas. Hoy, sin embargo, hay un número creciente de usuarios para la instalación de aplicaciones Web con limitadas habilidades y conocimientos. Hay miles de códigos abiertos de aplicaciones Web disponibles, pero no están normalizadas de una forma sencilla para los que van a instalarlas. Muchas de las aplicaciones Web más populares vienen con su propio programa de instalación, pero los usuarios aún están obligados a desempaquetar los ficheros de un archivo y visitar una dirección Web para ejecutar el programa de instalación. Esto puede ser simple para desarrolladores, pero para el usuario medio no lo es. Además, el programa de instalación casi siempre carece de los privilegios necesarios para realizar todas las operaciones y lograr así una instalación completa de la aplicación. Muchas veces, el programa de instalación pide al usuario cambiar los permisos sobre determinados archivos y carpetas. No sólo se trata de una petición

irritante, sino que además para muchos usuarios está más allá de sus conocimientos. El usuario no debe realizar tareas adicionales después de ejecutar el instalador, es decir, la propia herramienta debe realizar todas las operaciones necesarias para instalar la aplicación Web.

Una aplicación Web (WA) se define como una aplicación de *software* entregada a usuarios desde un servidor Web a través de una red como la *World Wide Web* (WWW) o una Intranet. Conceptualmente se puede encontrar también como una aplicación que requiere de un servidor Web para funcionar e interactuar con los usuarios vía navegador [9].

Servidor Web: Programa de *software* que se ejecuta como un servicio en un equipo que es responsable de brindar el contenido de la Web [10].

La forma común de instalar una aplicación Web en nuestros días consiste en seguir instrucciones que se brindan en la documentación de la misma. Muchas de estas aplicaciones incluyen un programa de instalación que facilita algunas de las tareas de instalación. Sin embargo, todavía hay una gran cantidad de trabajo manual que se debe hacer antes y después de usar una de estas herramientas.

En primer lugar, el usuario debe descargar el paquete de la aplicación, desempaquetarlo y seguir todas las orientaciones para la ejecución del mismo. Frecuentemente, un archivo de configuración debe ser modificado o creado antes de que el instalador pueda comenzar su ejecución.

Una vez concluida su labor, a menudo hay tareas adicionales que deben ser llevadas a cabo para completar el proceso de configuración. La mayoría de las veces, los permisos de directorios deben ser modificados para que el servidor Web pueda escribir archivos a estos directorios entre otras configuraciones.

1.6 ¿Cómo se instalan los productos desarrollados por la UCI con características similares al SIGEP?

En la Universidad de las Ciencias Informáticas (UCI) se desarrollan en estos momentos un conjunto de proyectos tanto del ámbito nacional como internacional. Estos proyectos solucionan problemas sociales estratégicos principalmente para los países de Cuba y la República Bolivariana de Venezuela. Algunos de estos, tales como: el proyecto Centro de Tratamiento y Análisis de Información de Seguridad Ciudadana (CTAISC) también conocido como 171, encargado de la gestión de los procesos policiales en Venezuela; el proyecto Identidad, el cual gestiona la identificación de los ciudadanos venezolanos; y el proyecto Aduana, para la gestión de los procesos aduanales cubanos, tienen puntos en común como la plataforma de desarrollo, la arquitectura y el cliente. Por ende tienen un conjunto de características de distribución y despliegue que se asemejan al SIGEP. Analizar el soporte que brindan

estas aplicaciones de *software* para los procesos de despliegue e instalación es una tarea prioritaria a realizar antes de brindar una propuesta de solución al problema que ocupa la investigación en curso.

Los proyectos Identidad y Aduana se encuentran actualmente en construcción y no presentan una herramienta de instalación, es decir, para las actividades referentes al proceso de despliegue no se tiene pensado una solución que realice un proceso automatizado y parametrizable.

Para el caso del proyecto 171 se tiene la aplicación dividida en dos componentes, una parte es una aplicación de escritorio y la otra una aplicación Web. Para el primer caso se cuenta con un instalador creado sobre la plataforma *.Net*, mientras que para la aplicación Web sobre la plataforma Java, no se dispone de una herramienta de instalación. Los procesos de configuración e instalación se hacen de forma manual en el servidor de aplicaciones.

1.7 Conclusiones

En el presente capítulo se puede concluir con que no se cuenta en la actualidad, con una herramienta de instalación o componente reutilizable adecuado para realizar la actividad de instalación de *software* del SIGEP. En los proyectos que se desarrollan en la Universidad de las Ciencias Informáticas no existe solución para esta problemática. Se propone la creación de un instalador de *software* cuyas herramientas y tecnologías de construcción sean: *IzPack* y *Apache Ant*.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se hace referencia a las características tanto de la arquitectura de despliegue del SIGEP como a la arquitectura de *software* utilizada. Además se detallan los problemas existentes en la instalación actual de esta aplicación informática, así como las características que debe tener esta actividad. Se definen los procesos correspondientes a las actividades del despliegue de un *software* y las entidades que intervienen en ellos, dando una descripción más detallada de los mismos para el SIGEP.

2.2 Arquitectura de despliegue del SIGEP

La distribución del SIGEP se concibe de la siguiente forma: un sistema instalado en cada uno de los centros penitenciarios e internados judiciales y para cada uno de estos centros un servidor de base de datos local que se actualiza regularmente de la base de datos central a través de mecanismos de réplica. Además se dispone del mismo sistema instalado en los Centros de Tratamiento Comunitario, las UTASP y las coordinaciones regionales, los cuales están conectados a la aplicación central, donde radican sus datos y la lógica de los procesos que ejecutan.

Para la comunicación entre cada uno de los centros penitenciarios se establecerá una infraestructura de comunicaciones que incluya conexiones WAN entre el centro de datos y los 30 puntos de establecimientos penitenciarios, 19 CTC, 33 UTASP y 5 coordinaciones regionales, distribuidos en todo el territorio nacional. Se dispondrá además de un canal de comunicación de alta velocidad entre la Dirección General y el centro de datos.

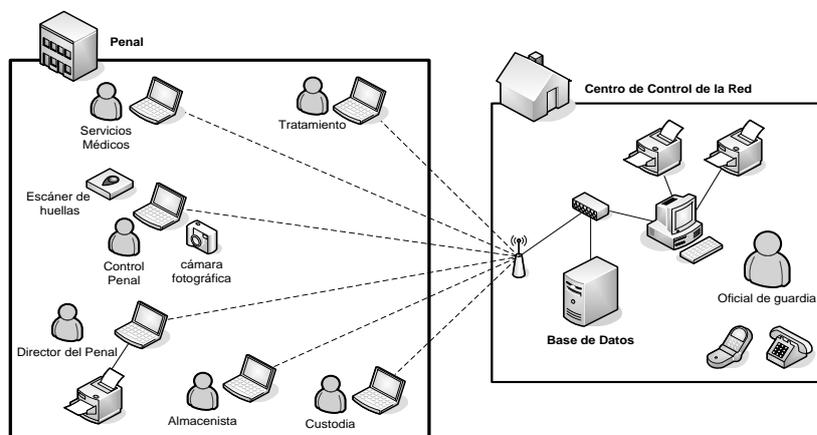


Figura 4 - Arquitectura de distribución del SIGEP

El centro de datos ubicado en las instalaciones del Ministerio del Interior y Justicia (MIJ) se destinará fundamentalmente para alojar las aplicaciones del SIGEP y del portal Web de la institución. Aprovechará la infraestructura previamente creada sobre la que se instalará la tecnología necesaria para garantizar los siguientes servicios: base de datos del SIGEP, servidor de aplicaciones, portal Web, gestor de contenidos, base de datos del portal Web, almacenamiento de datos, entre otros.

El SIGEP es un sistema basado en la arquitectura de tres capas, distribuidas cada una según las siguientes especificaciones:

Cliente: Computador de tecnología Intel. La aplicación se ejecuta a través de un navegador de Internet instalado sobre cualquier sistema operativo (se sugiere alguna distribución de *Linux* y *Windows XP* solo para las estaciones de trabajo a las que se conectarán los dispositivos externos: escáner de huellas y de documentos, cámara digital). Máquina Virtual de Java (JVM), versión 1.5 para aplicaciones clientes que la necesiten.

Servidor Web: Computador de tecnología Intel. En este servidor radica la lógica de negocio de la aplicación. Se utilizará el sistema operativo *Linux RedHat 4.0, Advanced Server*, para los servidores del centro de datos y alguna distribución de *Linux* para los servidores locales de los establecimientos penitenciarios. Para el servidor de aplicaciones se cuenta con *Apache TomCat* versión 5.x.x sobre la plataforma *Java Runtime Environment (JRE)*, versión 1.5.

Servidor de base de datos: Computador de tecnología Intel. Sistema operativo *Linux RedHat 4.0, Advanced Server*, para los servidores del centro de datos y alguna distribución de *Linux* para los servidores locales de los establecimientos penitenciarios. Servidor de base de datos *Oracle 10g Standard Edition*.

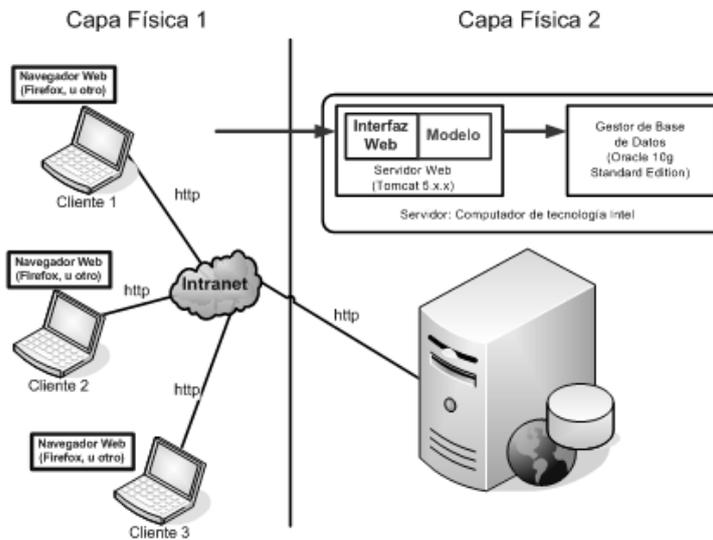


Figura 5 - Distribución de las capas de software

2.3 Arquitectura de software del SIGEP

La aplicación está dividida en tres capas lógicas fundamentales:

- **Capa de presentación:** En esta capa se encuentran las vistas y la lógica de presentación. En la lógica de presentación se maneja todo el flujo Web utilizando la implementación del patrón Modelo Vista Controlador (MVC) y las vistas son los recursos que le permiten al cliente visualizar la información, estos pueden ser páginas HTML¹², documentos en formato PDF, hojas de cálculo, entre otras.
- **Capa de servicios de negocio:** La presente capa encapsula toda la lógica de la aplicación en fachadas de negocio que son utilizadas por los controladores en la capa de presentación y se exponen algunos procesos de negocio a través de interfaces de servicios. A estas fachadas de negocio se le aplican la seguridad a nivel de métodos y de objetos de negocio, auditorías, cache, política de transacciones, entre otros.
- **Capa de acceso a datos:** Maneja los objetos de acceso a datos abstrayéndolos del mecanismo de persistencia usado, a través de interfaces que exponen las operaciones de persistencia definidas para cada uno de los DAOs y que son implementadas utilizando *Hibernate*, que es un *framework* de mapeo objeto-relacional (ORM).

¹² HTML, del inglés *HyperText Markup Language*: es el lenguaje de marcado predominante para la construcción de páginas Web.

2.4 Principales problemas de la instalación manual del SIGEP

En la actualidad la instalación del SIGEP se caracteriza por ser un conjunto de tareas que se realizan de forma manual. El arquitecto de *software*, responsable de la integración de cada una de las piezas de la versión a liberar del producto, genera y configura un componente WAR¹³ a partir del IDE¹⁴ Eclipse¹⁵, que contiene todo el código de la aplicación. Se introducen configuraciones específicas en los archivos de configuración para el centro en el que se está instalando el *software*, entendiéndose esto como: dirección de la base de datos, dirección física donde se guardarán los *logs* de auditoría del sistema, los subsistemas y módulos a desplegar, el prefijo del establecimiento, etc.

- Todos estos cambios que se realizan directamente sobre los ficheros de configuración contenidos en el WAR imponen que, para cada centro penitenciario se cree un nuevo componente, es decir, un nuevo WAR con tan solo las modificaciones y configuraciones particulares para el centro en el cual se instalará la aplicación.

El SIGEP está constituido por un conjunto de subsistemas y módulos estructurados según las áreas que conforman la actividad penitenciaria. Los principales subsistemas son: Control penal, Tratamiento, Salud, Administración del penal, Sala situacional, Seguridad y custodia, Administración penal, entre otros que actualmente se encuentran en desarrollo.

- El despliegue de estos subsistemas y sus módulos en el servidor de aplicaciones se realiza manualmente. Se adicionan o eliminan las referencias a estos directamente sobre el fichero de configuración *sigep.properties* en dependencia de si la instalación en el centro penitenciario es total o parcial.

Existen establecimientos penitenciarios que no contemplan dentro de sus actividades las funcionalidades de todos los subsistemas y módulos. Esto, unido a que algún módulo en específico no esté funcional en el momento de la instalación genera el siguiente problema:

- Se despliega en el servidor de aplicación el código de estos subsistemas, lo cual trae consigo que se copien archivos innecesarios en el servidor que representan funcionalidades de la aplicación que no se utilizarán.

Debido a la separación física de los componentes utilizados, en la actualidad la instalación del sistema se concibe de forma desacoplada. Tales componentes como: el servidor de aplicación *Apache TomCat*

¹³WAR, del inglés *Web Application Archives*: archivo utilizado en el empaquetamiento de una aplicación Web.

¹⁴IDE, del inglés *Integrated Development Environment*: programa compuesto por un conjunto de herramientas para un programador.

¹⁵Eclipse: IDE de código abierto independiente de una plataforma para desarrollar aplicaciones.

y la maquina virtual de Java no se encuentran en un mismo medio de almacenamiento (CD, memoria flash) lo cual trae consigo la siguiente dificultad:

- Los componentes anteriormente mencionados se instalan en el servidor del centro penitenciario de una forma desorganizada. No se validan correctamente las precondiciones que el sistema necesita para su correcta instalación. Las tareas del despliegue no fluyen de manera continua como un único proceso automatizado.

Otro problema actual en la instalación del SIGEP está relacionado con el personal encargado de dicha actividad.

- El personal seleccionado debe tener conocimientos sobre tecnologías J2EE, administración de aplicaciones Web, así como del sistema operativo *Linux*. Se hace indispensable la presencia en el establecimiento penitenciario de un personal técnico especializado en la arquitectura del SIGEP, preferentemente el arquitecto de *software* del equipo de desarrollo u otro miembro que domine perfectamente estas cuestiones.

2.5 Características que debe tener la instalación del *software* SIGEP

La instalación del SIGEP debe ser un proceso automatizado y parametrizable que facilite al usuario las configuraciones pertinentes. Para esto se debe contar con una herramienta que apoye y guíe todo el proceso de transferencia de los diferentes subsistemas y módulos para la gestión penitenciaria.

Precondiciones de la instalación

La herramienta debe verificar una serie de precondiciones antes de llevar a cabo la instalación ya que su activación depende del ambiente que debe estar previamente creado. Estas son:

- Verificar la existencia de la máquina virtual de Java con la versión recomendada o una superior que sea compatible, de no ser así instalarla.
- Verificar que esté instalado el servidor de aplicaciones *Web Apache TomCat*, así como su versión (5.x.x). Si no existe la versión deseada o una superior, proceder a instalar una compatible.

Parámetros configurables

La herramienta necesita una serie de parámetros de los cuales depende la configuración e instalación de la aplicación:

- Parámetros de configuración de la base de datos: Usuario, contraseña, dirección URL¹⁶ del servidor de base de datos. Esta última seccionada en: puerto, nombre del servidor y SID¹⁷ de la base de datos.
- Direcciones físicas de los log de auditoría: Camino dentro del sistema de archivos donde se almacenarán los ficheros que registran el comportamiento del sistema, acciones de los usuarios y los errores de la aplicación.
- La dirección física de la instalación del sistema: Ruta específica en el sistema de archivos donde será instalada la aplicación.
- Nombre del centro penitenciario: Identificador del centro penitenciario en el cual se desarrolla el despliegue.

2.6 Flujo de procesos de la instalación del SIGEP

De acuerdo a las actividades del proceso de despliegue de *software* descritas en el Capítulo # 1 se han definido un conjunto de procesos. Dentro de cada proceso intervienen entidades que interactúan entre sí convirtiéndose en las entradas y salidas de las tareas que conforman a estos procesos. La figura # 6 representa cómo interactúan de manera general estos procesos.

¹⁶ URL, del inglés *Uniform Resource Locator*: secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

¹⁷ SID, del inglés *System ID*, identificador interno que se utilizará para referenciar la base de datos.

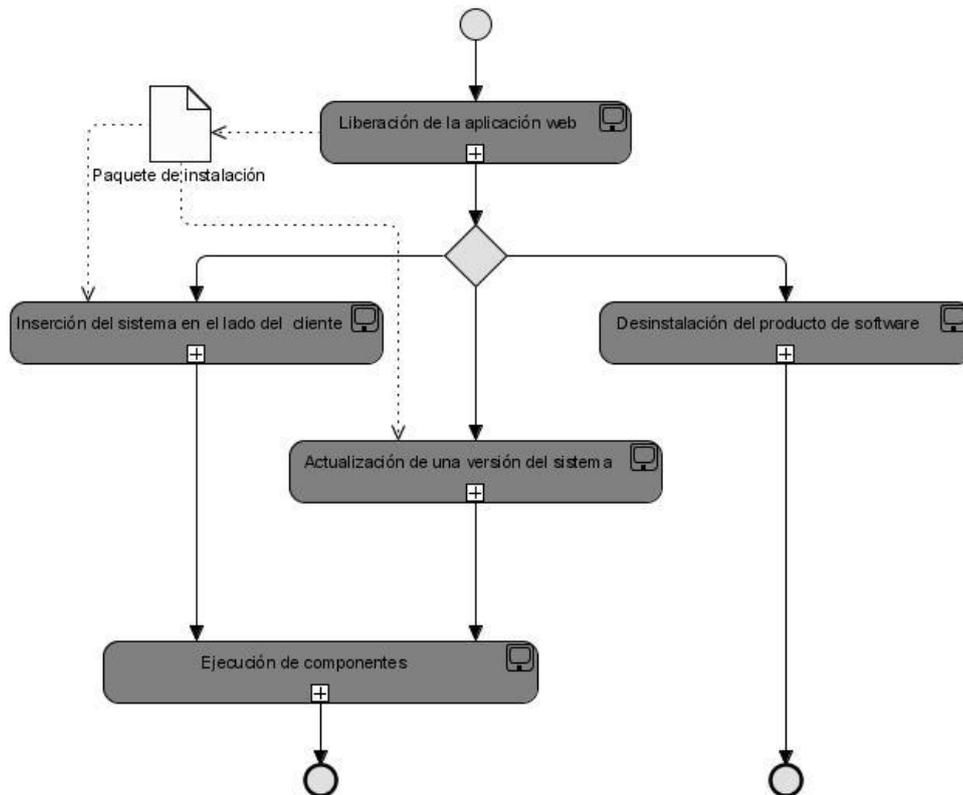


Figura 6 - Flujo de procesos de la instalación del SIGEP

Entidades comunes que se manejan en los procesos de la instalación del SIGEP.

Existen un conjunto de entidades que transitan a través de los procesos mostrados en la figura # 6. Todas se agrupan en una entidad general denominada *paquete de instalación*, la cual se describe a continuación:

Paquete de instalación: Medio de soporte en el cual van a estar agrupados los componentes de instalación del SIGEP en un sistema de archivos y carpetas. Su estructura está dada por los siguientes elementos:

- Versión estable del producto de *software*: Contiene los subsistemas de la aplicación empaquetados por módulos, así como los distintos archivos de configuración.
- Máquina virtual de Java: Plataforma sobre la cual se ejecutarán los componentes en el servidor.
- Servidor de aplicaciones: En este caso una versión superior a *Apache TomCat 5.0*.
- Manual de instalación del SIGEP: En este documento se explica cómo llevar a cabo la instalación del sistema operativo *Linux RedHat 4.0, Advanced Server*. De acuerdo a este sistema operativo se detallan los pasos a seguir para la instalación de la JVM y el *TomCat*.

Además contiene una guía para la navegación a través de los paneles de configuración de la instalación.

2.6.1 Liberación de la aplicación Web

Este proceso está centrado principalmente en la actividad de liberación descrita y representada en el Capítulo # 1 y consiste en la configuración y empaquetado de todos los componentes necesarios para que la aplicación se pueda ejecutar en el lado del cliente, además se copia el empaquetado resultante en algún dispositivo de almacenamiento ya sea un CD, memoria flash, etc. De esta manera puede ser transferido para su posterior instalación y configuración en el lado del cliente.

Principales entidades que intervienen

En el proceso de liberación de la aplicación Web se generan un conjunto de entidades de las cuales algunas se agrupan dentro de la entidad llamada *paquete de instalación* descrita anteriormente. Se definen otras entidades que son netamente de la liberación tales como:

- *Estructura lógica*: Modela la disposición de los archivos y carpetas para la liberación Web (figura # 7). Representa una descripción gráfica de la estructura de carpetas modelada. Su objetivo es guiar la creación y configuración de las entidades de *construcción e instalación*.

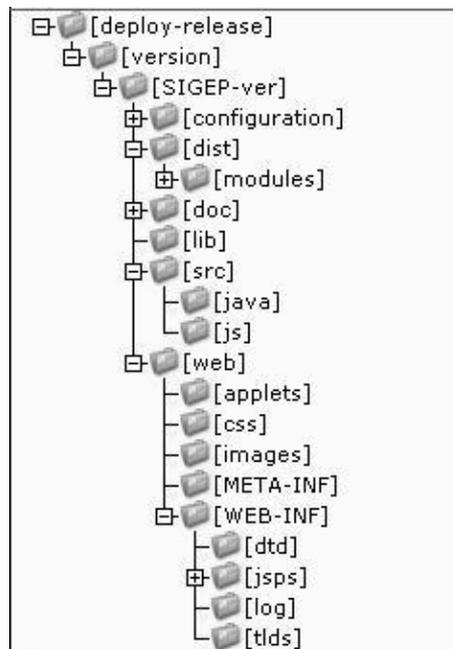


Figura 7 - Modelo lógico de la liberación Web

- *Archivo de construcción*: Archivo XML que define las tareas interpretadas por *Apache Ant* encargadas de generar todas las estructuras de ficheros y carpetas físicamente en este proceso. Contiene toda la información para transformar la arquitectura del sistema en la estructura física de la liberación Web. También tiene información e instrucciones para crear el empaquetado de los módulos de la aplicación. Agrega los componentes al paquete de instalación.
 - *Archivo de instalación*: Fichero en formato XML que presenta un conjunto de tareas interpretadas por *IzPack* destinadas a empaquetar la aplicación Web y crear el instalador de la aplicación.
 - *Archivo integrador*: Contiene la información para integrar los ficheros de *construcción* y de *instalación*. Hace que las instrucciones de estas entidades fluyan como parte de un mismo proceso automatizado, ejecutando las tareas requeridas de una de forma secuencial.
- Release-SIGEP*: Es la estructura física de carpetas y archivos que conforman la liberación de la aplicación Web.

Tareas que conforman la liberación Web

En la figura # 8 se muestra el flujo de las tareas que tienen como resultado la liberación, así como las entidades que intervienen en cada una de ellas.

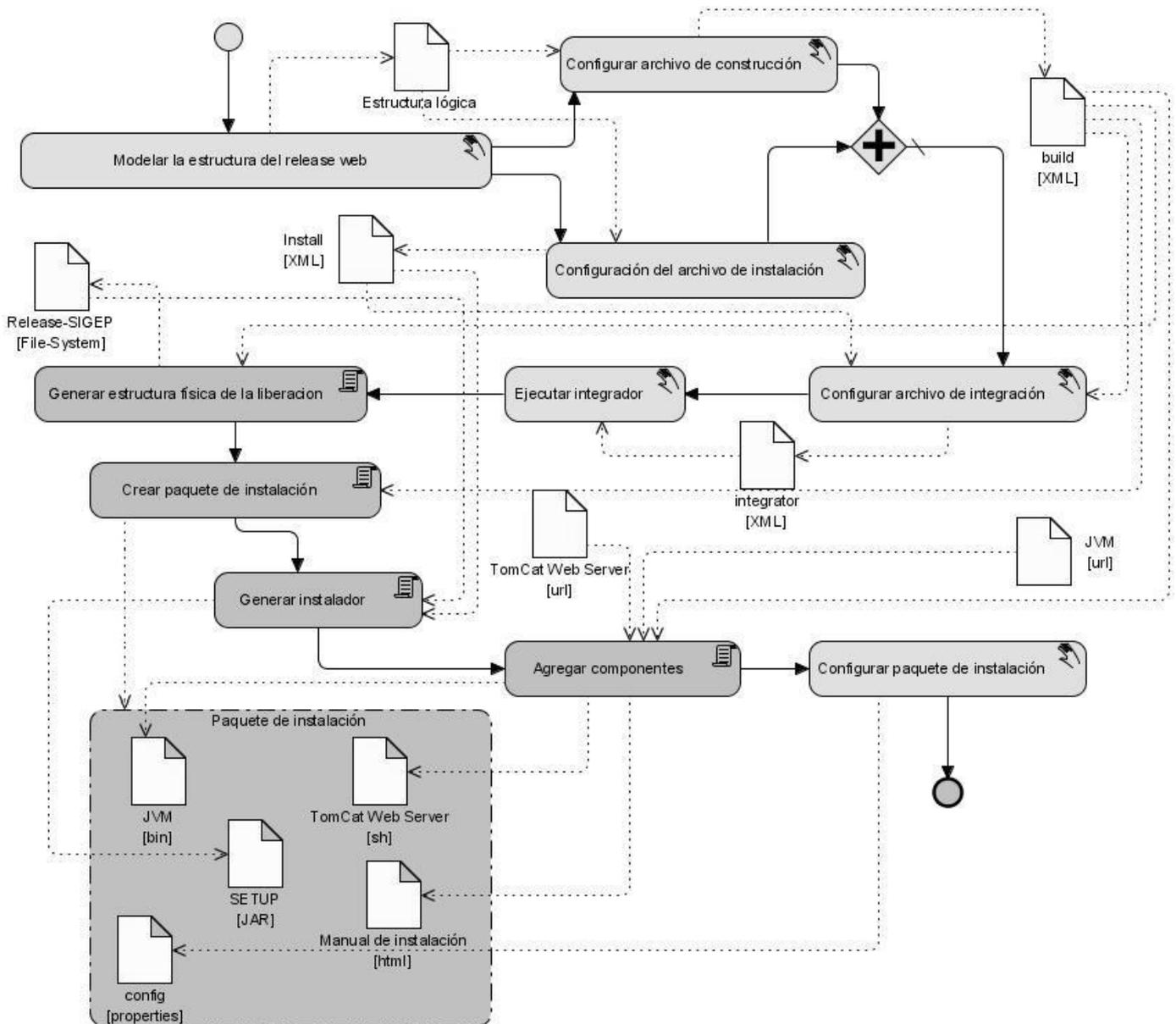


Figura 8 - Flujo de tareas para la liberación de la aplicación Web del SIGEP

A continuación se ofrece una descripción del proceso modelado en la figura # 8:

Se modela la estructura del sistema de archivos y carpetas con el propósito de obtener un diagrama que sirva de orientación en actividades posteriores. Tomando como entrada el modelo resultado de la actividad anterior se configuran los ficheros de instalación y de construcción. Esta actividad tiene como objetivo definir el conjunto de tareas para crear la estructura física de la liberación y crear el ejecutable de instalación. Luego se integran ambos archivos para que sus instrucciones fluyan como parte de un mismo proceso automatizado.

Luego de tener listo todo el ambiente, se ejecuta el integrador antes configurado y se desencadenan una serie de eventos de forma automática. Se genera la estructura física de la liberación y el paquete de instalación mediante las tareas definidas en el archivo de construcción y se genera el instalador utilizando el fichero de instalación. Una vez concluida la ejecución de estas tareas, se agregan los componentes al paquete de instalación, y se crea un fichero que tendrá la descripción del paquete de instalación, dando como resultado la liberación final de la aplicación Web.

2.6.2 Inserción del sistema en el lado del cliente

La inserción del sistema en el lado del cliente está enmarcada en la actividad de instalación del modelo general de despliegue (figura # 9). Consta de dos etapas fundamentales: la transferencia o entrega del producto desde el proveedor hasta el cliente y las operaciones de configuración en el servidor de aplicaciones. El *paquete de instalación* constituye la entidad principal que nutre dicho proceso. De una forma más específica, los elementos que componen a esta entidad tales como: el servidor de aplicaciones *Apache TomCat*, la JVM, y el manual de instalación, son las entradas para las tareas que describen a este proceso.

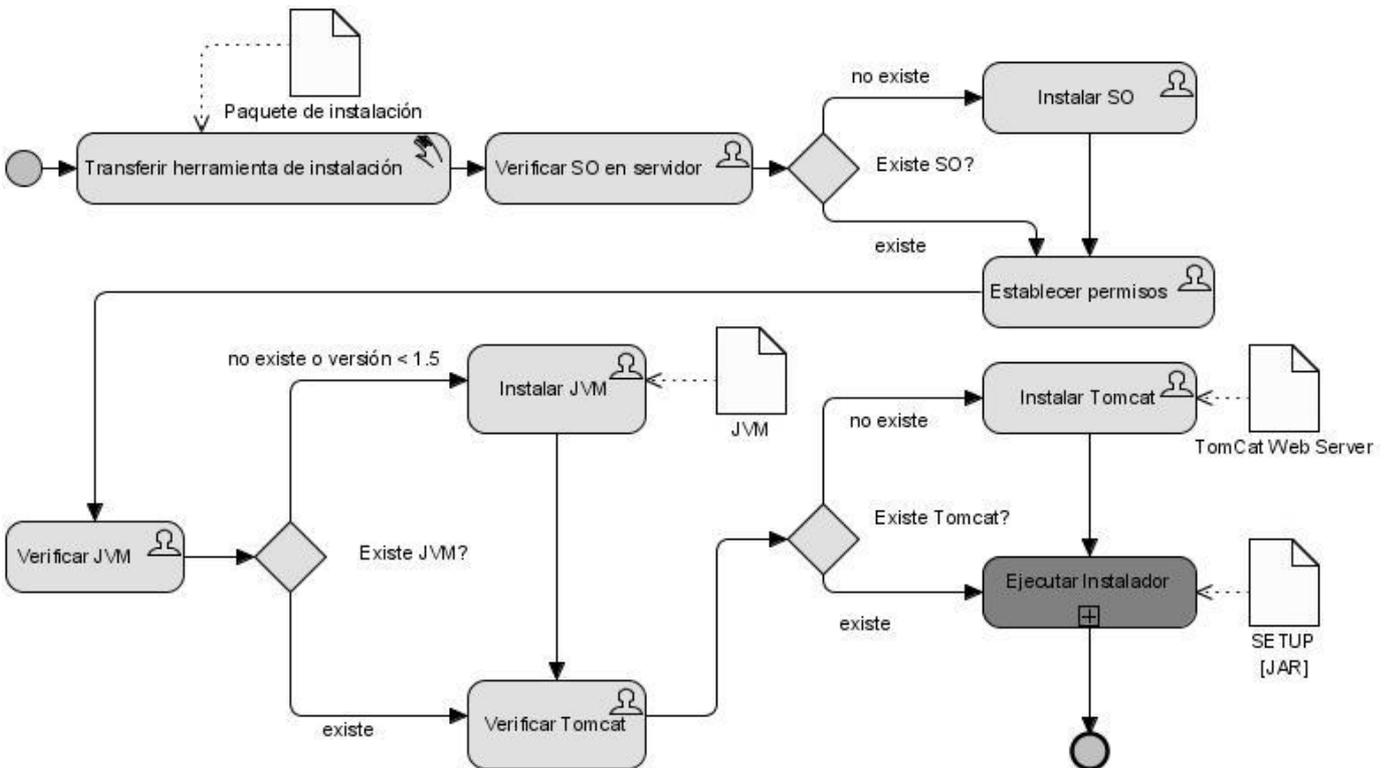


Figura 9- Flujo de tareas para la inserción del SIGEP en el lado del cliente

El flujo de tareas comienza con la transferencia de la herramienta hacia el lado del cliente. Esta tarea parte de la entrega del paquete de instalación a la entidad o persona responsable de la transferencia, que a su vez hace llegar el producto al establecimiento penitenciario. En este punto el administrador del sistema debe realizar las siguientes verificaciones y configuraciones:

- Verificar sistema operativo del servidor. En caso de no estar presente, se procede a la instalación del mismo. Si existe un sistema operativo este debe ser *Linux RedHat 4.0, Advanced Server*. La versión mencionada es la requerida pues constituye la versión con la cual el proveedor de software cuenta para el sistema operativo a utilizar.
- Establecer los permisos o privilegios necesarios en la ejecución de los elementos que componen el *paquete de instalación*.
- Verificar la existencia de la JVM. En caso de no estar en el servidor, se procede a la instalación de la misma. Si existe alguna versión, esta debe ser superior o igual a la versión 1.5. Solo así se garantiza que la aplicación funcione correctamente.
- Verificar la existencia del servidor Web *Apache TomCat*. Instalar en el caso de no estar, si existe, asegurar que la versión sea la indicada: superior a *Apache TomCat 5.0*.

Es necesario destacar que para cada instalación de los componentes mencionados anteriormente el administrador debe cumplir a cabalidad con los pasos incluidos en el manual de instalación. Solo de esta manera, se garantizará la calidad requerida en la integración de estos componentes.

Luego de tener una correcta configuración en el servidor, lo cual quiere decir que los elementos previamente configurados están en un estado funcional, el administrador del sistema puede proseguir con la ejecución de las tareas siguientes:

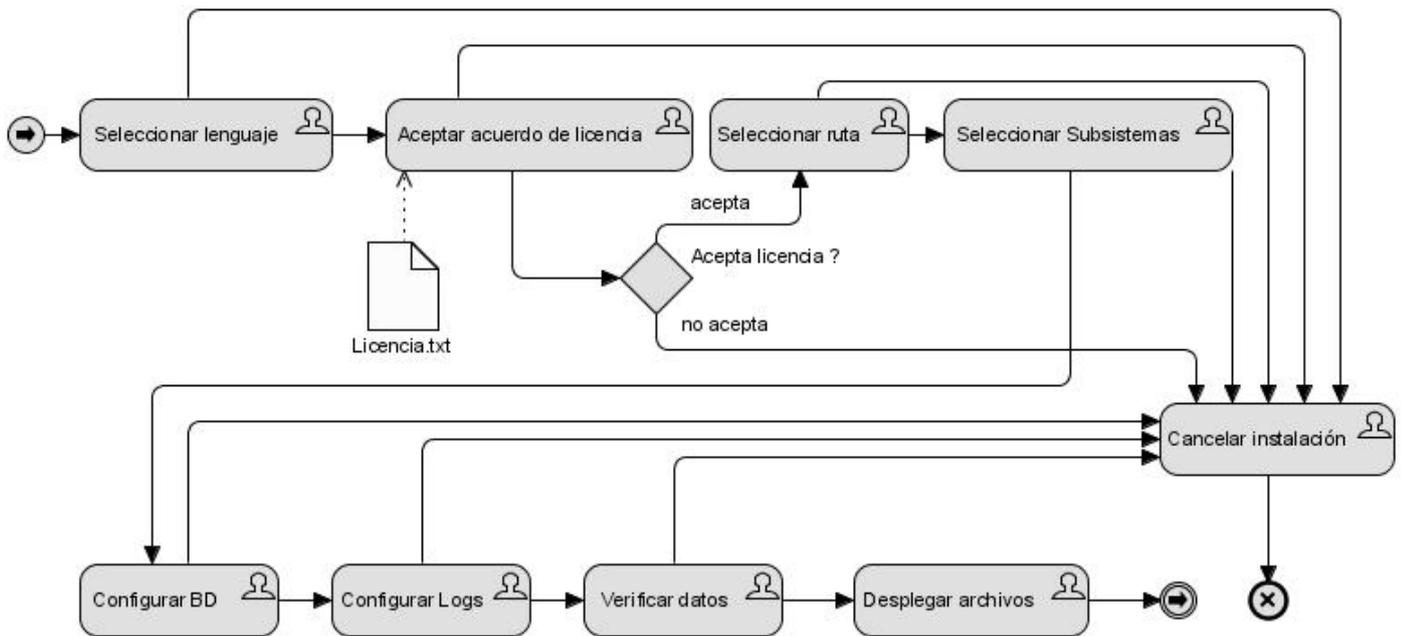


Figura 10 - Flujo de instalación de la aplicación

La herramienta de instalación constituye un *wizard*¹⁸ que guía al administrador del sistema a través de un conjunto de ventanas de información y entrada de datos. La primera tarea está dada por la selección del lenguaje. Todas las informaciones en lo adelante se mostrarán en el lenguaje seleccionado. Seguidamente se presentan ventanas de información que incluyen la bienvenida y un conjunto de especificaciones técnicas que instruirán al usuario sobre las características generales del proceso que se lleva a cabo, los requerimientos del sistema, los problemas que se pueden presentar, etc.

Con el objetivo de representar el contrato establecido entre los proveedores del *software* y el cliente, se muestra el acuerdo de licencia para el producto. La ventana que contiene a este, detendrá el flujo de actividades hasta que no se marque la opción de aceptación de licencia.

La configuración siguiente consiste en la selección de la ruta de instalación. Por defecto la ruta que debe mostrarse es la dirección donde está el servidor de aplicación *Apache TomCat* que dependerá de las variables de entorno y del sistema de archivos del sistema operativo, en este caso *Linux RedHat 4.0, Advanced Server*.

Una de las configuraciones más importantes constituye la selección de los subsistemas y módulos a desplegar en el servidor. Por defecto deben estar seleccionados todos los subsistemas para una

¹⁸ Wizard: aplicación que ayuda al usuario a ejecutar una tarea en forma eficaz.

instalación completa. Además se debe mostrar un paquete común deshabilitado que represente los archivos requeridos para la aplicación. Para el caso en que solo se requiere una parte de la aplicación, se procederá a reestructurar la selección.

Las configuraciones de la base de datos y de los ficheros que contienen las trazas de auditoría constituyen las siguientes tareas respectivamente. En el primero de los casos, se deben introducir datos que definen la conexión de la aplicación Web con la base de datos, tales como: usuario y contraseña para la autenticación, servidor, puerto y SID para la construcción de la URL. Además se introduce el identificador del establecimiento penitenciario en el cual se desarrolla el despliegue. Para el caso de los ficheros de auditoría se deberán seleccionar las direcciones físicas donde se guardarán estos tipos de ficheros definidos para el SIGEP, los cuales se describen a continuación:

- *Ficheros de auditoría*: Registran las trazas de la aplicación tales como: la llamada a métodos, eventos, peticiones al servidor, descripción de las peticiones, acciones de los usuarios, entre otras.
- *Ficheros de errores*: Registran todos los errores de la aplicación detectados durante la ejecución del sistema.
- *Ficheros de sistema*: Recopila información general así como las generadas en los *logs* de auditoría y de errores anteriormente mencionados.

Para una verificación de cada parámetro se muestra una ventana con un sumario de todas las configuraciones introducidas. Esto permite el cambio de cada entrada en caso de haber ocurrido algún error. Se procede entonces al despliegue de los archivos al servidor de aplicaciones, mostrando en una barra de progreso el estado de la instalación hasta completar la misma. Cabe destacar que durante este proceso se genera la herramienta de desinstalación para la aplicación instalada.

De manera inusual, pero con una posibilidad real, la ejecución anteriormente descrita puede ser cancelada si el administrador de sistema lo estima conveniente.

2.6.3 Ejecución de componentes

Este proceso es el encargado de garantizar que cada uno de los componentes que son requeridos para el funcionamiento de la aplicación estén en ejecución. Hace referencia a la actividad de activación del modelo general de despliegue del Capítulo # 1. Una vez que la aplicación se encuentra en el servidor de aplicaciones, solo restaría reiniciar a este último. La tarea constituiría un paso inherente a la terminación del *wizard* de instalación y se ejecutaría de forma automática preferiblemente a través

de un *script*. Una vez finalizada se puede decir que la aplicación está lista para su uso en las estaciones de trabajo del establecimiento penitenciario.

2.6.4 Desinstalación de la aplicación

Las tareas referentes a la retirada de la aplicación comienzan con la ejecución de la herramienta de desinstalación previamente generada en el proceso de inserción del sistema. Estas tareas se agrupan en dos secciones fundamentales:

Detener componentes en ejecución: Si es necesario se procede a la detención de cada uno de los componentes que el sistema utiliza durante su ejecución. Este proceso hace referencia a la actividad nombrada desactivación descrita en el modelo general de despliegue del Capítulo # 1.

- Eliminación de los archivos: Propio de la actividad de desinstalación (ver epígrafe 1.3), este proceso se encarga de la retirada de cada uno de los archivos inherentes a la aplicación de *software* que se desea eliminar.

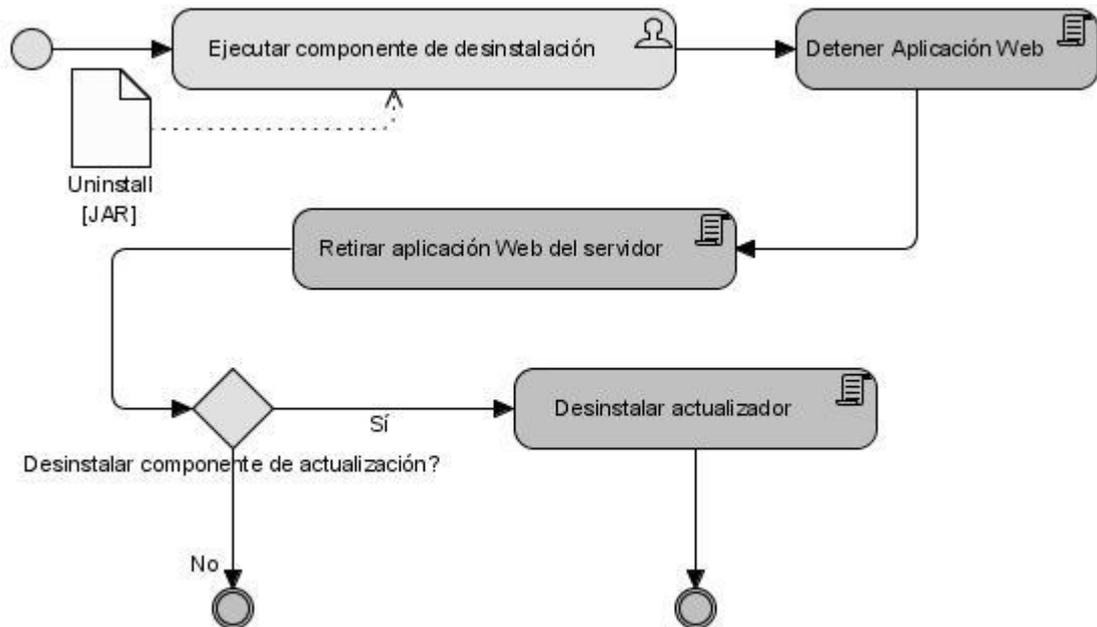


Figura 11 - Flujo de tareas de desinstalación de la aplicación

Existe una tarea que puede o no realizarse para este proceso en dependencia de las necesidades del usuario. Dicha tarea constituye la desinstalación de la herramienta de actualización.

2.6.5 Actualización de una versión del sistema

Un caso particular de la instalación constituye la actualización de un sistema. Para esta actividad, solo se copiarán en el servidor los archivos necesarios de una versión nueva del producto y no el sistema en su totalidad.

En este caso, la actualización de SIGEP se realiza a través de otra aplicación. Esta aplicación es un demonio que se encuentra en constante ejecución dentro del sistema operativo debido a que el proceso de actualización puede durar varios días en dependencia de la infraestructura de red y a la conectividad de la misma.

2.7 Conclusiones

En este capítulo se ha adaptado el modelo de despliegue de *software* estudiado en el Capítulo # 1 para dar solución a la problemática existente en el SIGEP, además se introdujo el flujo de actividades necesarias que la herramienta debe cumplir, así como las entidades identificadas en el mismo.

CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se expone cómo se ha implementado la herramienta de instalación propuesta en el Capítulo # 2. Se presenta una descripción sobre los elementos que se han construido para el desarrollo del instalador y cómo interactúan entre sí. Además se describe la arquitectura de la herramienta utilizada en su construcción así como la estructura de los principales ficheros de configuración creados con dicha herramienta. Finalmente se muestran los resultados obtenidos en la simulación de una instalación en un centro penitenciario.

3.2 Construcción de la liberación Web

Para automatizar el proceso de creación de la liberación Web descrito en el Capítulo # 2 se han creado un conjunto de archivos XML cuyo contenido son tareas de *Ant*. Estos ficheros se interrelacionan invocando las tareas que los integran.

3.2.1 Configuración del archivo de construcción

Este archivo contiene las secciones referentes a la creación física de la estructura de la liberación Web. Utiliza tareas de *Ant* para la creación, copia, modificación y eliminación de ficheros y carpetas que se encuentran en una versión estable de la aplicación (*mkdir*, *copy*, *delete*). Además se utilizan otros recursos como los `<fileset>`, `<exclude>` e `<include>` que usan patrones para la inclusión o exclusión de los archivos deseados. Para el caso del uso de los `<fileset>` dentro de la tarea `<copy>` un comportamiento por defecto sería la exclusión de los archivos que no son parte de la liberación, en este caso los subdirectorios `.svn`¹⁹.

```
<target name="make-release" description="make the release
directories structure">
  <mkdir dir="${target.deployment.dir}" />
  ...
  <copy todir="${target.deployment.dir}/configuration">
    <fileset
      dir="${src.dir}/configuration/${src.vnz.path}/configuration/">
      <exclude
        name="${src.dir}/configuration/${src.vnz.path}/configuration/basic.properties" />
      </fileset>
    </copy>
  ...
```

¹⁹ .svn: referente a los directorios asociados al software de sistema de control de versiones.

Además se definen secciones para el empaquetamiento de los módulos de cada subsistema del SIGEP en un formato de archivo Java (JAR²⁰), para ello se utilizan tareas de *Ant* especializadas en la creación y configuración de este tipo de fichero.

```
...
<jar destfile="${target.core.dir}/tratamiento/tratamiento-
trabajo.jar">
  <fileset dir="${src.classes.dir}">
    <include name="**/${src.vnz.path}/tratamiento/trabajo/**/*" />
  </fileset>
</jar>
...
```

Para establecer un único punto de partida de las configuraciones de estos ficheros se creó el archivo *release.properties*. La función de este consiste en exteriorizar los valores de los atributos de las tareas mencionadas anteriormente.

```
<property file="release.properties" />
```

3.3 IzPack como herramienta de construcción

Como se expuso en el Capítulo # 1, *IzPack* constituye la herramienta a utilizar en la construcción del instalador para el SIGEP. A continuación se presentan elementos tales como: la arquitectura del compilador, sus principales componentes, así como la estructura de los ficheros de configuración fundamentales que maneja dicha herramienta para el desarrollo de la solución propuesta.

3.3.1 Arquitectura de *IzPack*

El sistema de compilación interno de *IzPack* es muy modular. Se puede usar de dos maneras fundamentales:

- Desde la línea de comando del sistema operativo utilizado.
- A partir de la herramienta de generación de código *Apache Ant*.

²⁰ JAR, del inglés *Java Archive*: tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java

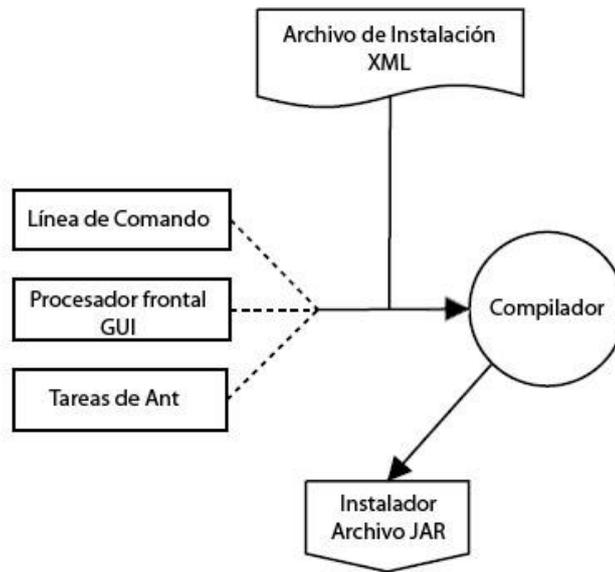


Figura 12 - Arquitectura de compilación de *IzPack*

El compilador toma como entrada un fichero de instalación XML el cual define las características del instalador a crear. Este fichero, definido como el elemento fundamental de configuración, contendrá información detallada de la aplicación a instalar, los paneles GUI que se utilizarán, así como los recursos que sean necesarios cargar para cada panel especificado [6].

El compilador puede generar distintos tipos de instaladores:

- Instaladores estándares: Archivo simple que responde a las características de un instalador convencional.
- Instaladores a través de la Web: Los paquetes de archivos se encuentran en un servidor http y son descargados en tiempo de ejecución.

Esta información no se encuentra en el XML de instalación, sino que forma parte de los parámetros que se le agregan en el momento de su ejecución. En la siguiente instrucción se ejemplifica con el uso del parámetro *-k* para el caso específico del SIGEP.

```
"..\IzPack\bin\compile" ./install.xml -b . -o SIGEP.jar -k Standard
```

3.3.2 Configuración del archivo de instalación

IzPack define para cada instalador un conjunto de componentes. Las preferencias gráficas, paquetes, paneles, recursos y los lenguajes son los elementos fundamentales. Cada uno de estos se traduce a configuraciones específicas dentro del fichero XML de instalación. En la figura # 12 se muestra gráficamente la disposición de esos componentes.



Figura 13 - Estructura de componentes de *IzPack*

La estructura general para el fichero de instalación denominado *install.xml* quedaría de la siguiente forma:

```

<installation version="1.0">
  <info> (...)</info>
  <guiprefs> (...)</guiprefs>
  <locale> (...)</locale>
  <resources> (...)</resources>
  <panels> (...)</panels>
  <packs> (...)</packs>
</installation>
  
```

En esta estructura el elemento raíz lo constituye la etiqueta *<installation>* la cual solo tiene el atributo *version* que define la versión del esquema XML para el fichero de instalación y es usado por el

compilador de *IzPack* para verificar la compatibilidad con el archivo XML. Todas las declaraciones y configuraciones a realizar serán dentro de esta etiqueta.

Información general

El elemento `<info>` contiene información general del instalador. El nombre, la versión, los autores, la URL del sitio o portal Web oficial de la aplicación a instalar, las configuraciones para generar una herramienta de desinstalación entre otros tales como la mínima versión de la JVM requerida son los aspectos más importantes a destacar de esta sección.

```
<info>
  <appname>@{info.appname}</appname>
  <appversion>@{info.appversion}</appversion>
  <appsubpath>@{info.appsubpath}</appsubpath>
  <authors>
    <author name="@{info.author}" email="" />
  </authors>
  <javaversion>@{info.javaversion}</javaversion>
  <uninstaller name="@{info.uninstal.name}" write="yes" />
  <url>@{info.url}</url>
</info>
```

Los valores para cada una de estos parámetros se encuentran exteriorizados en el fichero *install-config.properties*. La idea central de la existencia de este fichero es facilitar la escritura de estos valores. Con una estructura clave-valor se evita navegar sobre el XML de instalación, por lo que las configuraciones se hacen mucho más sencillas para el desarrollador. La declaración dentro del *install.xml* que hace referencia a este fichero quedaría de la siguiente forma:

```
<properties>
  <property file="config/install-config.properties"/>
</properties>
```

Especificaciones de los componentes GUI

Para las especificaciones de los componentes GUI del instalador se define la etiqueta `<guiprefs>` cuyos elementos fundamentales son las propiedades visuales para los paneles de instalación tales como las dimensiones (largo y ancho), la alineación de los componentes, colores, tamaños y estilos para los textos, la presencia o no de una barra de progreso, etc. La declaración para esta sección quedaría de la siguiente forma:

```

<guiprefs width="@{gui.width}" height="@{gui.height}"
resizable="@{gui.resizable}">
    <modifier value="true" key="showDebugWindow"/>
    <modifier key="labelGap" value="2"/>
    <modifier key="headingLineCount" value="1"/>
    <modifier key="headingPanelCounter" value="progressbar"/>
    <modifier key="headingPanelCounterPos"
value="inNavigationPanel"/>
<laf name="metouia">
    <os family="unix" />
</laf>
</guiprefs>

```

De igual manera que en el epígrafe anterior, los valores para dichas etiquetas se encuentran formando parte del archivo de configuración *install-config.properties*.

Internacionalización

Para el soporte a múltiples lenguajes en la etiqueta *<locale>* se definen el conjunto de lenguajes que podrán ser seleccionados por el usuario final. Dentro de esta etiqueta cada lenguaje se especifica mediante la norma ISO3²¹. Para el caso del instalador de SIGEP solo son de interés los lenguajes español (spa) e inglés (eng) debido a las características y a la naturaleza del cliente. Su declaración sería como se muestra a continuación:

```

<locale>
    <langpack iso3="spa" />
    <langpack iso3="eng" />
</locale>

```

Con respecto a este tema existen un conjunto de archivos XML que se han creado para la internacionalización de cada texto mostrado al usuario en la herramienta de instalación. Estos se dividen en dos grupos fundamentales:

- Ficheros para el lenguaje español.
- Ficheros para el lenguaje inglés.

Básicamente existen los mismos ficheros para cada lenguaje. La diferencia radica en el nombre del archivo al cual se le agregan los códigos ISO: *_eng* o *_esp* en dependencia del lenguaje seleccionado;

²¹ISO3: Lenguaje de código para definir idiomas.

y a los valores de las claves para cada mensaje a mostrar. A continuación se ejemplifica dicho comportamiento para cada lenguaje:

Para el caso del idioma inglés:

```
<langpack>
  <!-- messages -->
  <str id="common.description" txt="All of its common"/>
  ...
</langpack>
```

En el caso del idioma español:

```
<langpack>
  <!-- mensajes -->
  <str id="common.description" txt="Todo lo que es común"/>
  ...
</langpack>
```

Para los archivos de información y licencia se tendría de la misma forma, un fichero para cada lenguaje donde solo varía su contenido en dependencia del lenguaje a mostrar.

Paneles

En el elemento `<panels>` se le especifican al compilador de *IzPack* cuáles son los paneles que tendrá el instalador. Estos irán apareciendo durante el transcurso de la instalación de acuerdo al orden en que estén declarados en la sección correspondiente. Estos paneles pueden estar ya predefinidos o se pueden crear nuevos en dependencia de las necesidades de la solución propuesta.

El instalador del SIGEP está constituido por un total de 10 paneles que conforman el *wizard* de instalación. El primero de estos paneles es el denominado "*HelloPanel*". Este da la bienvenida al usuario mostrando el nombre, la versión, los autores y la dirección Web de la aplicación.

```
<panels>
  <panel classname="HelloPanel" />
  <panel classname="InfoPanel" />
  <panel classname="LicencePanel" />
  <panel classname="TargetPanel" />
  <panel classname="TreePacksPanel" />
  <panel classname="UserInputPanel" />
  <panel classname="UserInputPanel" />
  <panel classname="XInfoPanel" />
  <panel classname="SummaryPanel" />
  <panel classname="InstallPanel" />
```

```
<panel classname="SimpleFinishPanel" />
</panels>
```

Seguidamente se encuentran los paneles “*InfoPanel*” y “*LicencePanel*” los cuales dan la posibilidad de mostrar al usuario final los tradicionales archivos de información y licencia respectivamente. Destacar que estos ficheros normalmente son archivos de texto estático (extensión txt) y se pueden sustituir por documentos HTML con tan solo cambiar los paneles a “*HTMLInfoPanel*” y “*HTMLLicencePanel*”. En este caso, en la sección de recursos se podrían definir imágenes entre otros elementos para esta variante.

El panel denominado “*TargetPanel*” tiene como objetivo permitirle al usuario la selección de la ruta de instalación. Para el caso del SIGEP, se tiene una aplicación Web la cual se desplegará sobre el servidor de aplicaciones *Apache TomCat*. De acuerdo a esto se muestra en el panel la ruta de instalación de este servidor de aplicaciones. Esta es generada a partir de la variable de entorno CATALINA_HOME, definida para el sistema operativo del servidor.

Seguidamente se declaran una serie de paneles que controlan la configuración y entrada de parámetros del usuario. El panel “*TreePacksPanel*” el cual define la selección de los paquetes a instalar de una forma jerárquica, para SIGEP subsistema y módulos. En otro panel “*UserInputPanel*”, se permite la entrada de configuraciones de la base de datos y la selección de la dirección donde se guardaran los ficheros que contienen las trazas de la aplicación en el servidor.

Para mostrar la información que ha sido registrada durante todo el proceso de instalación se tienen los paneles “*XInfoPanel*” y “*SummaryPanel*”. Ellos mostrarán todas las entradas anteriormente introducidas por el usuario dando la posibilidad de rectificarlas antes de comenzar la copia de los ficheros al servidor. Se agrupan en dos secciones, en una primera parte se exponen las configuraciones de la base de datos y los *log* de auditoría mientras que en la segunda se muestra la ruta de instalación y los subsistemas seleccionados con sus correspondientes módulos.

El panel “*InstallPanel*” mostrará como ocurre el despliegue de la aplicación a través de una barra de procesos. Si no existe ningún error durante este proceso se dará una confirmación al usuario de que la instalación ha concluido por medio del panel *SimpleFinishPanel*.

Recursos

La mayoría de los paneles anteriormente descritos requieren de datos adicionales para ejecutar sus tareas. Estos datos son suministrados a dichos paneles en forma de recursos. Los principales recursos utilizados son imágenes, las especificaciones de los componentes gráficos para el caso del

“*UserInputPanel*”, los archivos de información y licencia para el caso de los paneles “*InfoPanel*” y “*LicencePanel*” y los recursos de internacionalización que contienen cada uno de los mensajes a mostrar en dependencia del lenguaje seleccionado. La estructura de esta sección quedaría de la siguiente forma:

```
<resources>
  <res id="TargetPanel.dir" src="config/tomcatPath.txt" />
  <!-- Imágenes -->
  <res id="installer.langsel.img" src="img/logodgsp.PNG" />
  <res id="Installer.image" src="img/logodgsp.PNG" />

  <!-- Recurso para paneles de entrada de datos-->
  <res id="userInputSpec.xml" src="config/userInputSpec.xml" />

  <!-- Recursos para paneles de información -->
  <res id="LicencePanel.licence" src="lang/spa/Licencia.txt" />
  <res id="LicencePanel.licence_eng" src="lang/eng/Licence.txt"
/>
  <res id="InfoPanel.info" src="lang/spa/Leeme.txt" />
  <res id="InfoPanel.info_eng" src="lang/eng/Readme.txt" />
  <res id="XInfoPanel.info" src="lang/spa/info.txt" />
  <res id="XInfoPanel.info_eng" src="lang/eng/info.txt" />

  <!-- Recursos para paneles de internacionalización-->
  <res id="packsLang.xml_eng" src="lang/eng/packsLang.xml_eng" />
  <res id="packsLang.xml_spa" src="lang/spa/packsLang.xml_spa" />
  <res id="userInputLang.xml_eng"
src="lang/eng/userInputLang.xml_eng" />
</resources>
```

Paquetes

El elemento `<packs>` contiene el conjunto de archivos a desplegar por el instalador. Está dividido en dos secciones fundamentales, una sección para la declaración de todos los archivos comunes de la aplicación y otra para la declaración de los subsistemas y módulos del SIGEP. A continuación se muestra un ejemplo que contiene las dos secciones principales para esta etiqueta:

```
<packs>
<pack name="common" required="yes" id="common">
  <!-- Archivos del common-->
  <file src="@{release.path}/dist/modules/common"
targetdir="$INSTALL_PATH/WEB-INF/lib" />
  ...
  <!-- Archivos de los subsistemas-->
```

```

<pack name="Datos personales" required="no" id="datospersonales"
parent="controlpenal">
  <description>datospersonales</description>
<filesrc="@{release.path}/dist/modules/controlpenal/controlpenal-
datospersonales.jar" targetdir="$INSTALL_PATH/WEB-INF/lib" />

  </pack>
...
</packs>

```

Dentro de cada etiqueta `<pack>` se indican los ficheros que contendrán variables a sustituir. Dichas variables registrarán los valores de las entradas de usuario en los distintos archivos de configuración de la aplicación. También se especifican los ficheros *scripts* que se quieran ejecutar inmediatamente después que termine la instalación o la desinstalación de la aplicación.

```

<pack name="common" required="yes" id="common">
<!-- Archivos que contienen variables a sustituir -->
  <parsable targetfile="$INSTALL_PATH/WEB-
INF/classes/vnz/sigep/configuration/hibernate.properties"
type="javaprop"/>
  ...
  <!-- Archivos a ejecutar-->
  <executable targetfile="$INSTALL_PATH/execute/restarTomCat"
stage="postinstall"/>
</pack>

```

3.4 Integración de componentes

Tanto el archivo de instalación como el de construcción previamente descritos se integran a través del fichero *integrator.xml*. El objetivo principal de este último constituye la creación de un proceso automatizado. Este está estructurado en dos tipos de secciones atendiendo a su función: las secciones encargadas de crear la estructura de liberación y las que empaquetan los módulos de la aplicación.

- Ejemplo de las secciones de construcción:

```

<target name="deploy-relese">
  <ant antfile="${basedir}/structures.xml" target="make-
release" />
</target>
...

```

- Ejemplo de las secciones de empaquetado:

```

<target name="merge-L1">
  <ant antfile="${base.dir}/build(jar).xml" target="L1" />
</target>

```

Las secciones anteriormente mencionadas son agrupadas en una que se encarga de definir la secuencia lógica en que deben ser invocadas.

```
<target name="integrate" depends="deploy-release, merge-L1,  
merge-L2, merge-L3, installer-package"/>
```

Al igual que el fichero de construcción, se utilizan las configuraciones exteriorizadas en el archivo *release.properties*. De esa forma se asegura que tengan una misma línea de configuración, mucho más accesible y de rápida manipulación.

3.5 Simulación del entorno real. Resultado obtenido

Para demostrar la eficiencia del modelo de instalación creado para el SIGEP en la actual investigación, así como las potencialidades de la herramienta desarrollada, se han sentado las bases en uno de los laboratorios productivos donde radica el equipo de desarrollo del SIGEP. Se simularon las condiciones físicas de los establecimientos penitenciarios venezolanos tratando de ajustar esas condiciones a la realidad de estos centros.

Primeramente dada una versión estable en el repositorio que maneja el control de versiones se creó el paquete de instalación. Se instaló el mismo sistema operativo de los servidores de los establecimientos penitenciarios venezolanos, en este caso, *Linux RedHat 4.0, Advanced Server*. Se instalaron además la jre1.6 de Java así como el servidor de aplicaciones *Apache TomCat 5.5*. Además se habilitó en el propio servidor, el gestor de base de datos definido para la solución de *software, Oracle 10g Standard Edition* con una base de datos de igual estructura a la que actualmente presentan los centros penitenciarios.

Una vez creadas las condiciones en el servidor, se procedió a la instalación de la aplicación del SIGEP. La ejecución de dicha tarea se llevó a cabo siguiendo los pasos definidos en el Anexo # 1.

3.5.1 Resultado obtenido

Los procedimientos descritos anteriormente dieron a la investigación en curso los siguientes resultados:

- La liberación de la aplicación Web, de un conjunto de actividades manuales, pasó a ser un proceso automatizado.
- La herramienta de instalación se integró al sistema de archivos del sistema operativo utilizado.

- La lectura y escritura en tiempo de ejecución sobre los ficheros de configuración de la aplicación fue satisfactoria.
- La transferencia hacia el servidor de todos los archivos que componen a la aplicación se realizó de forma íntegra.
- El procedimiento de instalación del SIGEP se mostró de una forma más amigable para el usuario en comparación con el usado en los establecimientos penitenciarios hasta la fecha.

3.6 Conclusiones

En este capítulo se expusieron los elementos fundamentales que fueron creados dado el flujo de procesos descrito para el SIGEP. Se pudo apreciar cómo las tecnologías utilizadas tales como *IzPack* y *Apache Ant* dieron solución a las tareas propuestas. Además, con la elaboración de los ficheros de construcción, instalación e integración se automatizaron tres actividades fundamentales del despliegue del software: Liberación, Instalación y Desinstalación. Una vez creado el paquete de instalación, se probaron sus componentes principales los cuales dieron resultados satisfactorios.

CONCLUSIONES

Dado el estudio presentado en esta investigación se obtuvo como resultado un mecanismo de automatización para la liberación Web y la herramienta de instalación del Sistema de Gestión Penitenciaria venezolano, los cuales se integran perfectamente.

Los principales beneficios brindados por la solución propuesta fueron los siguientes:

- Obtención de la liberación Web del SIGEP de forma completamente automática.
- Interfaz gráfica de la herramienta de instalación.
- El tiempo de las actividades del proceso de despliegue disminuyó considerablemente comparado con el método que se utiliza actualmente en los establecimientos penitenciarios venezolanos.
- El nivel técnico requerido del personal encargado de la instalación del sistema informático es mínimo dado las características de la herramienta desarrollada.

Por lo anteriormente expuesto se puede arribar a la conclusión de que el objetivo de este trabajo se ha cumplido con la realización de dicha herramienta.

RECOMENDACIONES

- Estudiar la viabilidad de aplicar el mecanismo propuesto en otros proyectos de la Universidad.
- Agregar nuevas funcionalidades al mecanismo tales como la ejecución del *script* de la base de datos de la aplicación así como una prueba de conectividad a esta.
- Utilizar la solución como herramienta de instalación dentro del proceso de despliegue del SIGEP.

BIBLIOGRAFÍA

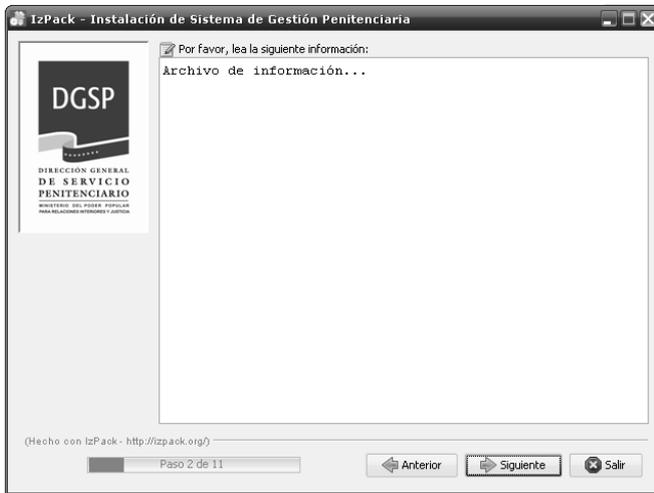
1. Richard S. Hall, Dennis Heimbigner, Alexander L. Wolf, A Cooperative Approach to Support Software Deployment Using the Software Dock, 1998.
2. A.Carzaniga, A. Fuggetta, R.S. Hall, A. van der Hoek, D. Heimbigner, A.L. Wolf. A Characterization Framework for Software Deployment Technologies. Technical Report CUCS-857-98, Dept. of Computer Science, University of Colorado, April 1998.
3. Antonio Carzaniga, A Characterization of the Software Deployment Process and a Survey of Related Technologies, September 1997.
4. Software Engineering Standards Committee of the IEEE Computer Society, IEEE Standard for Developing Software Life Cycle Processes, December 1997.
5. Arturo C. Arias Orizondo, Proyecto Técnico de Asesoría Especializada, Colaboración Médica Odontológica, Comunicación Institucional y Solución Tecnológica para apoyar la modernización del Sistema Penitenciario de la República Bolivariana de Venezuela, Caracas, Agosto 2006.
6. Julien Ponge, Elmar Grom, Fabrice Mirabile, Tino Schwarze, Klaus Bartz, IzPack Documentation.
7. Installer, WGMAC Glossary. Disponible en: <http://www.intbuild.com/WGGlossary.html>.
8. Package, WINE HQ. Disponible en: <http://www.winehq.org/site/docs/wineusr-guide/glossary>.
9. Web application, Simply. Disponible en: <http://www.simply.com.au/glossary.php?alphabet=W>.
10. Web server, What is The Leading IT Encyclopedia and Learning Center. Disponible en: http://whatis.techtarget.com/definition/0,sid9_gci213606,00.html.

ANEXOS

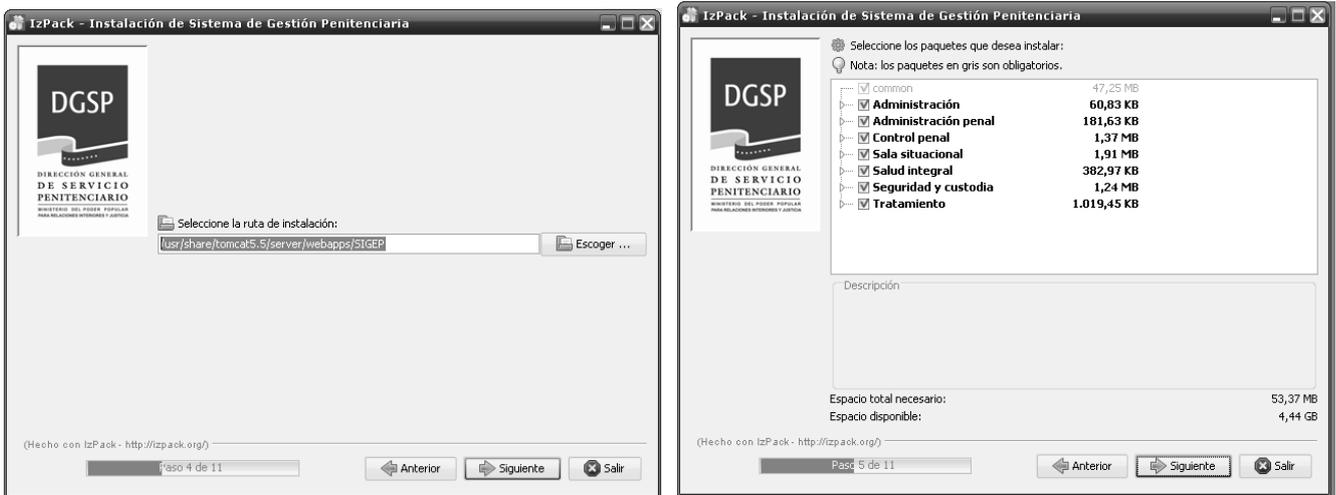
Anexo # 1 Ventanas de información y configuración de la herramienta de instalación del SIGEP



A la izquierda se expone la ventana de selección del lenguaje. A la derecha, la pantalla de bienvenida de la herramienta de instalación.



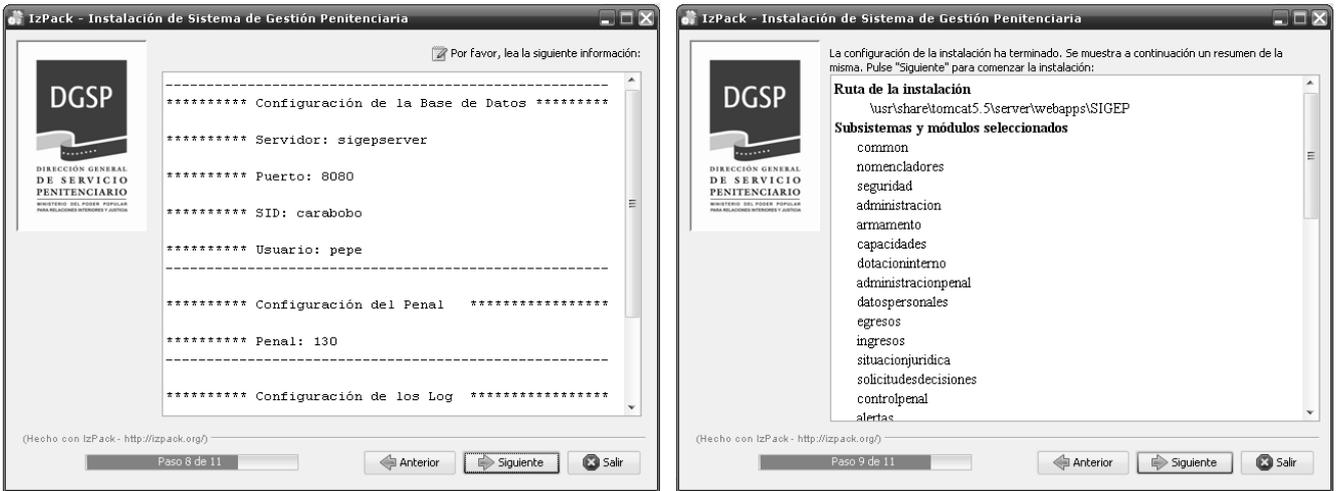
Las dos ventanas anteriores visualizan el contenido de los ficheros de información y licencia respectivamente.



En estas ventanas se ingresa la dirección donde se instalará el SIGEP, además se seleccionan los subsistemas y módulos deseados para el centro penitenciario.



A la izquierda se muestran las configuraciones que son necesarias para la interacción con la base de datos. A la derecha los campos que registran las direcciones físicas para guardar los ficheros que contienen las trazas de auditoría de la aplicación.



Las dos ventanas anteriores muestran todas las configuraciones previamente seleccionadas por el usuario.



Estas son las pantallas finales de la herramienta. La primera muestra un seguimiento de la transferencia de ficheros al servidor a través de una barra de progreso. Para el segundo caso, se presenta al usuario la finalización de las tareas y la dirección física donde se colocó la herramienta de desinstalación.

GLOSARIO

1. .svn: referente a los directorios asociados al software de sistema de control de versiones.
2. API, del inglés Application Programming Interface: es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
3. demonio (en inglés daemon, Disk And Execution Monitor): tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).
4. DLL, del inglés Dynamic Linking Library: término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.
5. docbook: dialecto de SGML (Standard Generalized Markup Language) que permite la escritura de documentación técnica.
6. Eclipse: IDE de código abierto independiente de una plataforma para desarrollar aplicaciones.
7. HTML, del inglés HyperText Markup Language: es el lenguaje de marcado predominante para la construcción de páginas Web.
8. IDE, del inglés Integrated Development Environment: programa compuesto por un conjunto de herramientas para un programador.
9. IEEE, del inglés Institute of Electrical and Electronics Engineers: asociación técnico-profesional mundial dedicada a la estandarización en las ciencias de la computación, telecomunicaciones, tecnologías biomédicas y otras ramas de las ciencias.
10. ISO3: Lenguaje de código para definir idiomas.
11. J2EE, del inglés Java 2 Platform, Enterprise Edition: Tecnología para desarrollar aplicaciones empresariales basadas en la Web.
12. JAR, del inglés Java Archive: tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java
13. JVM, del inglés Java Virtual Machine: Intérprete para el lenguaje de máquina de Java.
14. MSI, del inglés Microsoft Installer: los paquetes MSI sirven para poder instalar aplicaciones directamente en el equipo sin pedir interacción alguna por parte del usuario.

15. Open source: término en inglés con el que se conoce al software distribuido y desarrollado libremente.
16. Plugin: aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.
17. SID, del inglés System ID, identificador interno que se utilizará para referenciar la base de datos.
18. URL, del inglés Uniform Resource Locator: secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
19. WAR, del inglés Web Application Archives: archivo utilizado en el empaquetamiento de una aplicación Web.
20. Wizard: aplicación que ayuda al usuario a ejecutar una tarea en forma eficaz.
21. XML, del inglés Extensible Markup Language: metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

