



**ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**INGENIERÍA INFORMÁTICA**

**Curso Académico 2012/2013**

**Proyecto de Fin de Carrera**

**Generador de gestores de instalación  
para portales de descargas**

**Autora: Sara Lorenzo García**

**Tutor: Alberto Fernández Gil**

## RESUMEN

La descarga de software a través de Internet está en auge en la última década. Existen multitud de portales de descargas que ofrecen diferentes programas y que facilitan el acceso a estos programas a los usuarios.

Estos portales ofrecen un gestor de instalación con el software que el usuario quiere obtener. En este gestor además ofrecen diferentes ofertas que el usuario podrá decidir si instalar o no en su equipo.

La finalidad de este proyecto es la creación de un generador de gestores de instalación que sea capaz de generar estos instaladores para la plataforma Windows, en función a unos parámetros, para poder distribuirlos a través de un portal de descargas.

El generador creado recibirá unos parámetros y datos a través del enlace de descarga que aparecerá en el portal. A partir de estos datos se generará un instalador que será devuelto al usuario para su ejecución.

Este instalador estará formado por diferentes páginas que permitirán al usuario instalarse diferentes ofertas. Estas páginas son: página de bienvenida, página de toolbar, página/s de otras ofertas, página/s de ofertas a pantalla completa, página de descarga y página final. Algunas de estas páginas pueden mostrarse o no dependiendo de las opciones definidas por cada portal de descargas.

Al mismo tiempo el instalador será capaz de enviar estadísticas informando sobre los diferentes eventos que ocurren en el instalador, de esta forma estas estadísticas se podrán estudiar para determinar cuáles de los productos que se ofrecen son más populares, cuantas descargas finalizan con éxito, etc. Los eventos de los que se informan son la apertura del instalador, aceptación o rechazo de la instalación de una oferta, cierre del instalador, descarga del software,...

# ÍNDICE

<b>1. INTRODUCCIÓN</b> .....	<b>6</b>
1.1. CREACIÓN DE INSTALADORES: HERRAMIENTAS Y LENGUAJES .....	7
<b>2. OBJETIVOS</b> .....	<b>14</b>
2.1. DESCRIPCIÓN DEL PROBLEMA.....	14
2.2. METODOLOGÍA.....	15
<b>3. DESCRIPCIÓN INFORMÁTICA</b> .....	<b>17</b>
3.1. REQUISITOS .....	17
3.2. ANÁLISIS .....	19
3.2.1. LENGUAJE NSIS .....	20
3.2.2. USO DE TEMPLATES CON SMARTY .....	24
3.2.3. PHP Y HTML.....	26
3.3. DISEÑO.....	27
3.3.1. AÑADIR NUEVO PORTAL DE DESCARGAS .....	28
3.3.2. AÑADIR NUEVA OFERTA.....	28
3.3.3. AÑADIR NUEVO SOFTWARE .....	29
3.3.4. DISEÑO DE PANTALLAS Y BOTONES.....	30
3.4. IMPLEMENTACIÓN .....	38
3.4.1. PRIMERA FASE: INSTALADOR BÁSICO.....	38
3.4.2. SEGUNDA FASE: INSTALADOR CON OFERTAS .....	39
3.4.3. TERCERA FASE: FIRMA DEL INSTALADOR.....	47
3.4.4. CUARTA FASE: DESARROLLO WEB .....	48
3.4.5. ESQUEMA DEL FUNCIONAMIENTO FINAL.....	49
3.4.6. HERRAMIENTAS UTILIZADAS .....	51
3.5. EVALUACIÓN.....	52
3.5.1. EJEMPLO DE FUNCIONAMIENTO .....	52
3.5.2. EJEMPLO DE INSTALADOR EN ÁRABE.....	63
3.5.3. EJEMPLO DE INSTALADOR EN JAPONÉS .....	65
<b>4. CONCLUSIONES Y LÍNEAS FUTURAS</b> .....	<b>67</b>
4.1. LOGROS PRINCIPALES ALCANZADOS .....	67
4.2. LÍNEAS FUTURAS.....	68
<b>5. REFERENCIAS</b> .....	<b>69</b>
<b>ANEXO A</b> .....	<b>70</b>

## ÍNDICE DE FIGURAS

Figura 1. Tabla herramientas para la creación de gestores de instalación.....	13
Figura 2. Declaración de páginas .....	22
Figura 3. Declaración de páginas con el UI Moderno.....	22
Figura 4. Declaración de sección.....	23
Figura 5. Declaración de sección de desinstalación.....	23
Figura 6. Declaración de funciones.....	24
Figura 7. Declaración y uso de variables .....	24
Figura 8. Smarty combinado con NSIS .....	26
Figura 9. Diagrama de secuencia .....	27
Figura 10. Diagrama de secuencia para añadir un nuevo portal .....	28
Figura 11. Diagrama de secuencia para añadir una nueva oferta .....	29
Figura 12. Diagrama de secuencia para añadir una nueva oferta en un portal de descargas.....	29
Figura 13. Diagrama de secuencia para añadir un nuevo software.....	30
Figura 14. Página de bienvenida .....	31
Figura 15. Página de Toolbar .....	32
Figura 16. Página de otras ofertas.....	33
Figura 17. Página de ofertas a pantalla completa .....	34
Figura 18. Página de descarga .....	35
Figura 19. Página final.....	36
Figura 20. Botones .....	37
Figura 21. Fichero de idioma .....	39
Figura 22. Array de ofertas tipo toolbar .....	40
Figura 23. Array de ofertas tipo cupón .....	40
Figura 24. Array de otras ofertas.....	40
Figura 25. Ejemplo de función set_vars .....	41
Figura 26. Ejemplo fichero offers.inc.....	42
Figura 27. Ejemplo función install .....	45
Figura 28. Ejemplo fichero down.php de Dinar.....	49
Figura 29. Esquema de generación de un gestor de instalación .....	50
Figura 30. Portal de descargas .....	53
Figura 31. Firma digital .....	53
Figura 32. Página de Bienvenida .....	54
Figura 33. Log de la acción StartedInstall .....	54
Figura 34. Página de Toolbar .....	55
Figura 35. Log de la acción OfferShown .....	56
Figura 36. Log de la acción OfferAccepted .....	56
Figura 37. Log de la acción OfferInstallCompleted.....	56
Figura 38. Log de la acción OfferRejected .....	56
Figura 39. Página de otras ofertas.....	57
Figura 40. Log de la acción OfferShown en una misma página .....	57
Figura 41. Página de ofertas a pantalla completa .....	58
Figura 42. Página de descarga .....	59
Figura 43. Log de la acción StartDownload.....	59
Figura 44. Página final.....	60
Figura 45. Ejecución del programa .....	61
Figura 46. Log de la acción EndInstall .....	61
Figura 47. Salir del gestor de instalación .....	62
Figura 48. Log de la acción SkipAll .....	63
Figura 49. Log de la acción SkipThis .....	63

Figura 50. Página de bienvenida en árabe.....	63
Figura 51. Página de toolbar en árabe.....	63
Figura 52. Página de otras ofertas en árabe .....	64
Figura 53. Página de oferta a pantalla completa en árabe .....	64
Figura 54. Página de descarga en árabe .....	64
Figura 55. Página final en árabe .....	64
Figura 56. Página de bienvenida en japonés .....	65
Figura 57. Página de toolbar en japonés.....	65
Figura 58. Página de otras ofertas en japonés.....	65
Figura 59. Página de oferta a pantalla completa en japonés .....	65
Figura 60. Página de descarga en japonés.....	66
Figura 61. Página final en japonés.....	66

# 1.INTRODUCCIÓN

Actualmente, los ordenadores han cobrado una gran importancia en nuestra vida. Son, de hecho, parte fundamental de nuestra civilización actual y pieza fundamental en la gran mayoría de empresas y hogares. La explosión de la informática ha hecho que cualquier persona tenga acceso a un ordenador con gran facilidad y que se beneficie de sus grandes ventajas. Al mismo tiempo, el uso de Internet tiene un gran impacto en el mundo laboral, el ocio y el conocimiento a nivel mundial. Gracias a la red, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información.

El hecho de que la informática esté al alcance de cualquier persona, independientemente de sus conocimientos, ha hecho que existan numerosas aplicaciones informáticas que permiten a los usuarios realizar uno o diversos tipos de trabajo. Estas aplicaciones o programas son una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, o la gestión de un almacén.

La existencia de multitud de programas informáticos ha hecho que su distribución haya cambiado desde el comienzo de su explotación hasta lo que conocemos hoy en día. Hace unos años, los programas informáticos se vendían en tiendas con su CD de instalación y sus instrucciones de uso. Hoy en día, puedes adquirir cualquier tipo de programa a través de la web. Esta evolución ha ocasionado el nacimiento de numerosos portales de descarga que permiten al usuario encontrar y disfrutar del software que necesitan para mejorar su vida, así como opiniones de los mismos y versiones de prueba.

Cuando un usuario selecciona un programa para descargar de uno de estos portales de descarga, se descargará un gestor de instalación. Este gestor le permitirá descargarse el software que el usuario ha seleccionado, pero al mismo tiempo se le ofrecen otros productos para que los pueda instalar en su ordenador. El usuario puede decidir si los instala o no y al finalizar el proceso ya tendrá descargado en su equipo el programa seleccionado.

## 1.1. CREACIÓN DE INSTALADORES: HERRAMIENTAS Y LENGUAJES

El rápido crecimiento de la distribución de software ha propiciado, al mismo tiempo, un aumento de las herramientas y lenguajes para la creación de gestores de instalación. Actualmente existen estas herramientas con sus correspondientes lenguajes para cualquier sistema operativo. A continuación se enumeran algunos ejemplos para los sistemas operativos Windows, Linux y Mac OS.

- **WINDOWS**

- Inno Setup<sup>1</sup>

Inno Setup es una aplicación gratuita que te permite crear tus propios gestores de instalación para Windows. Es perfecto para desarrolladores que crean sus propias aplicaciones y después quieren distribuirlas con un sencillo instalador en un único archivo .exe toda la aplicación final.

Además, Inno Setup soporta todas las versiones actuales de Windows, por lo que cubre cualquier necesidad que puedas tener.

Otras características destacadas de esta herramienta son:

- Permite crear tipos de instalaciones personalizadas, por ejemplo *Completa, Mínima o Personalizada*.
- Ofrece un desinstalador completo.
- Tiene soporte extensivo para instalación de aplicaciones de 64 bits.
- Tiene habilidad de comparar las versiones de un archivo, reemplazar archivos en uso, registrar librerías, DLLs e instalar fuentes.

---

<sup>1</sup> <http://www.jrsoftware.org/isinfo.php>

- Permite la creación de accesos directos en cualquier ubicación.
- Permite la creación de claves de registro y ficheros .INI.
- Soporte para instalaciones multilingüe-
- Soporte para instalaciones con contraseña y encriptadas.
- Soporte para instalaciones y desinstalaciones silenciosas.

– InstallShield<sup>2</sup>

InstallShield es una herramienta de software para crear gestores de instalación o paquetes de software de cualquier programa que se desarrolle. El software ofrece todo lo necesario para crear un excelente instalador con todas las funciones necesarias.

Sus características principales son:

- Dispone de un potente entorno de desarrollo con un esquema visible de ficheros, aplicaciones de ayuda que te ahorran mucho tiempo, programas integrados de línea de comando y plantillas para terceros componentes.
- Posee un editor de sintaxis resaltado.
- Se puede usar para administrar aplicaciones y paquetes software en una amplia gama de dispositivos móviles.
- Tiene soporte para 29 idiomas.
- Existe una versión para Linux.

– Clickteam Install Creator<sup>3</sup>

Clickteam Install Creator te permite crear sistemas de instalación para tus programas de forma rápida y muy sencilla, llevando todo el proceso paso a paso a través de un asistente. Estos pasos comprenden la selección de ficheros que incluye tu programa, la elección del directorio

---

<sup>2</sup> <http://www.installshield.com/>

<sup>3</sup> <http://www.clickteam.com/website/world/install-creator>

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 te permite personalizar hasta el último detalle de tu instalación, desde los textos y logos que aparecen hasta la inclusión de opciones adicionales, como registro de componentes DLLs o creación de iconos de acceso directo.

– BitRock<sup>4</sup>

BitRock es una herramienta que permite crear gestores de instalación para aplicaciones tanto en Windows, Linux y MacOs.

El gestor de instalación generado tendrá la apariencia del sistema operativo para el que habrá sido creado, es decir que se integrará a la perfección en su entorno gráfico y funcional sin necesidad de dependencias. Además, brinda soporte para 16 idiomas.

– NSIS (Nullsoft Install System)<sup>5</sup>

Nullsoft Install System es un sistema de instalación bastante pequeño, eficiente y muy capaz para todos los sistemas basados en Windows.

Está basado en secuencias de comandos y te permite crear la lógica para manejar incluso las más complejas tareas de instalación. Muchos plug-ins y secuencias de comandos ya están disponibles: puedes crear gestores de instalación web, comunicar con Windows y otros componentes de software, instalar o actualizar los componentes compartidos y mucho más.

Algunas de sus características más destacadas son:

---

4 <http://installbuilder.bitrock.com/>

5 [http://nsis.sourceforge.net/Main\\_Page](http://nsis.sourceforge.net/Main_Page)

- Genera instaladores de ejecutables independientes.
- Permite desinstalar (instalador puede generar un desinstalador).
- Interfaz de usuario personalizable (diálogos, fuentes, fondos, iconos, texto, marcas de comprobación, imágenes, etc).
- Completamente multilingüe, admite varios idiomas (incluyendo lenguajes RTL) en un instalador.
- Sistema de página: Puedes agregar páginas de asistente estándar o páginas personalizadas.
- Múltiples configuraciones de instalación (normalmente mínima, típica, completa) y configuración personalizada.
- Capacidad para mostrar un contrato de licencia en texto o formato RTF.
- Capacidad para detectar el directorio de destino desde el registro.
- Fácil de usar sistema de plug-in (muchos plug-ins para la creación de cuadros de diálogo personalizados, conexiones a Internet, descarga HTTP, parchado de archivos, llamadas a la API de Win32, etc. se incluyen)

- **LINUX**

- [Ubucompiler](#)<sup>6</sup>

Se trata de una herramienta gráfica que permite crear, configurar e instalar paquetes .deb partiendo de ficheros fuente. Con esta herramienta se puede ahorrar mucho tiempo ya que no será necesario recurrir a la consola de comandos para compilar los archivos fuentes y así poder instalarlos.

Es una herramienta muy simple en la que únicamente se debe indicar el directorio donde se encuentran los ficheros fuente para que

---

6 <https://code.google.com/p/ubucompiler/>

Cree un fichero .deb.

– Debian Package Maker<sup>7</sup>

Debian Package Maker es una herramienta ideal para crear paquetes Debian sin necesidad de comandos en el terminal, de forma gráfica y con todas las opciones necesarias.

Incluye los campos habituales que puedes encontrar en cualquier fichero .deb, como nombre del paquete, versión y descripción, así como una lista de las dependencias que necesitará para poder funcionar y los conflictos que puede tener con otros paquetes.

Además, se pueden añadir los script necesarios para automatizar procesos y así dejar listo el programa a instalar en el ordenador.

– BitRock

Como se ha descrito anteriormente BitRock también es capaz de crear gestores de instalación en la plataforma Linux.

– InstallShield

Existe una versión de esta herramienta que permite la creación de gestores de instalación en la plataforma Linux.

• **MAC OS**

– Installer Vise<sup>8</sup>

Installer Vise es una potente utilidad para crear gestores de instalación. Tiene una interfaz muy bien diseñada y docena de asistentes, que permiten crear con facilidad desde el instalador más

---

<sup>7</sup> <https://code.google.com/p/debianpackagemaker/>

<sup>8</sup> [http://www.mindvision.com/vise\\_x.asp](http://www.mindvision.com/vise_x.asp)

simple hasta el más complejo. Esto incluye evaluar el entorno operativo, búsqueda de ficheros existentes, mover, copiar o renombrar ficheros.

Una de las opciones más interesantes de Installer Vise es la habilidad para recrear las instalaciones de otros programas. Tan sólo debemos mantener activado Installer Vise en segundo plano mientras se realiza la instalación, de la cual obtendremos una captura con todos los cambios realizados.

Otras características de esta herramienta son:

- Posee un potente motor de compresión.
- Pantallas de instalación configurables.
- Acciones predefinidas: añadir entradas al Registro, ejecutar aplicaciones o crear accesos directos.
- Visualización de contratos de licencia.
- Instalación de fuentes.
- Soporte para todas las versiones de Mac.

– FileStorm<sup>9</sup>

FileStorm es una potente herramienta para la creación de instaladores para aplicaciones Mac OS. Funciona con total facilidad, con tan solo arrastras y soltar los ficheros en la ventana de FileStorm, ya tendrás tu instalador listo, con la posibilidad de personalizar la apariencia de las imágenes creadas, el diseño, la ubicación de los iconos, etc.

– BitRock

Además de permitir la creación de instaladores para las plataformas Windows y Linux, BitRock es capaz de crear instaladores para Mac OS.

---

<sup>9</sup> <http://www.mindvision.com/filestorm.asp>

A continuación se muestra una tabla de clasificación de estas herramientas por sistema operativo.

	<b>WINDOWS</b>	<b>LINUX</b>	<b>MAC OS</b>
<b>Inno Setup</b>			
<b>InstallShield</b>			
<b>Clicteam Install Creator</b>			
<b>BitRock</b>			
<b>NSIS</b>			
<b>Ubucompiler</b>			
<b>Debian Package Maker</b>			
<b>Installer Vise</b>			
<b>FileStorm</b>			

Figura 1. Tabla herramientas para la creación de gestores de instalación

## **2.OBJETIVOS**

### **2.1. DESCRIPCIÓN DEL PROBLEMA**

El propósito general de este proyecto es la creación de un generador de gestores de instalación totalmente parametrizable que permita al usuario obtener el software deseado al mismo tiempo que se le ofrecen diferentes productos para instalar en su equipo.

Una vez explicado el funcionamiento del generador de instaladores se van a enumerar los objetivos que se pretenden llevar a cabo en el desarrollo del presente proyecto.

1. El principal objetivo es crear un sistema de calidad, robusto y flexible que permita generar instaladores de forma transparente al usuario. Para ello, se va a realizar un generador que pueda ser ejecutado de manera web y que redirija el ejecutable generado al usuario que realizó la petición.
2. Por otro lado, el instalador generado deberá ser un sistema sencillo, intuitivo y fácil de manejar para el usuario final.
3. El generador de gestores deberá ser multilingüe para poder llegar al mayor número de usuarios. De este modo creará instaladores en diferentes idiomas y las ofertas deberán estar traducidas al idioma del instalador, pero en el caso de que alguna oferta no lo estuviera, ésta se mostrará en inglés.
4. Otro de los objetivos es enviar estadísticas de la ejecución de los instaladores para posibles estudios de viabilidad y mejora. Así se podrá saber el número de instaladores ejecutados, la cantidad de instaladores que no terminaron su ejecución, el número de ofertas instaladas, ...

## 2.2. METODOLOGÍA

Para el desarrollo de este software se ha seguido un modelo de proceso de desarrollo similar al *modelo incremental* [4]. Este modelo combina elementos del modelo lineal secuencial (aplicados repetitivamente) con la filosofía interactiva de construcción de prototipos. En este modelo, primero se crea un prototipo inicial (núcleo) que contiene requisitos básicos. Se revisa este prototipo inicial y se desarrolla un plan para el siguiente incremento, decidiendo qué funcionalidades añadir. El plan afronta la modificación del producto central a fin de cumplir mejor las necesidades del cliente y la entrega de funciones, y características adicionales. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo.

La metodología seguida nos permitía obtener una versión inicial que fuese funcional para poder evaluarla y después ir añadiendo mejoras. Como incremento más significativo se puede destacar la creación de un instalador base, el cual será usado para la generación de instaladores con las mismas características para reducir el tiempo de generación del mismo.

Durante el desarrollo del programa se han seguido una serie de fases que son comunes en la fabricación del software: captura de requisitos, análisis, diseño, implementación, pruebas y producción.

En la fase de captura de requisitos se estudiaron las distintas versiones de instaladores que existen en el mercado y se establecieron las principales características que tenía que cumplir el generador de gestores de instalación. Así mismo se decidieron cuales serían los productos que se ofrecerían al usuario, determinando que serían 3 tipos: toolbars (barra de herramientas), cupones y otros productos.

En la fase de análisis se decidieron los diferentes lenguajes de programación que se podían utilizar para la creación de los instaladores. Se decidió que el lenguaje a utilizar sería NSIS (Nullsoft Scriptable Install System), pero debido a sus limitaciones se concluyó utilizarlo combinado con Smarty.

En la fase de diseño se procedió a la creación de las plantillas de Smarty para poder llevar a cabo los requisitos explicados anteriormente y las estadísticas que se debían enviar para permitir un estudio de instaladores, de esta forma se podrá saber cuántos instaladores se han ejecutado, cuantos instaladores no llegaron a finalizar, ofertas instaladas, etc.

En la fase de implementación se crearon los scripts de NSIS y de las plantillas Smarty para la creación de los instaladores en base a los requerimientos establecidos en la fase de captura de requisitos. Y en las pruebas se realizaron los tests necesarios para comprobar el correcto funcionamiento del generador y su adecuación a los requisitos previos.

Por último, en la fase de producción se ha creado un portal de descargas sencillo que permite a un usuario descargarse instaladores de software con distintas ofertas y características. Durante esta fase se fueron corrigiendo pequeñas anomalías que fueron surgiendo.

## **3.DESCRIPCIÓN INFORMÁTICA**

A continuación se presenta el desarrollo realizado explicando cada una de las fases que han tenido lugar, como son: la captura de requisitos, el análisis de la aplicación, el desarrollo de la solución, su implementación y las pruebas realizadas.

### **3.1. REQUISITOS**

Después de realizar un estudio exhaustivo sobre los más importantes portales de descarga que existen, se han especificado las siguientes características fundamentales del software.

- El generador de gestores de descarga será capaz de generar un instalador de acuerdo a unos parámetros de entrada.
- El generador deberá crear instaladores en diferentes idiomas. Los idiomas elegidos son: árabe, chino, danés, alemán, holandés, inglés, francés, griego, italiano, japonés, coreano, noruego, portugués, español y sueco.
- El gestor de instalación generado deberá poderse ejecutar en cualquier versión de la plataforma Windows.
- El instalador deberá mostrar el nombre y una imagen identificativa del programa final. Si no tiene imagen, se mostrará una por defecto.
- El gestor de instalación creado podrá tener diferentes páginas:
  - Página de bienvenida: esta página mostrará una imagen y el nombre del software que se va a descargar el usuario y una licencia que deberá aceptar para continuar con la instalación.

- Página de toolbar: se ofrecerá al usuario la instalación de una barra de herramientas determinada, en función a las propiedades del instalador.
  - Página/s de otras ofertas: podrán ser una o dos páginas que mostrarán una lista de otros productos y que el usuario podrá decidir si se instala o no mediante checks.
  - Página/s de ofertas de pantalla completa: muestran ofertas a pantalla completa para que el usuario tenga más información sobre el producto.
  - Página de descarga: página donde se descarga el software deseado.
  - Página final: muestra información del producto descargado y los agradecimientos al usuario por usar el instalador.
- 
- Antes de ofrecer una oferta se comprobará si el equipo del usuario ya la tiene instalada para no volvérsela a ofrecer y así poder mostrarle otra nueva.
  - Las instalaciones de las ofertas que el usuario acepte se llevarán a cabo de forma paralela entre si y de manera que no interfieran en el funcionamiento general del instalador.
  - El instalador solamente podrá ofrecer como máximo una oferta de tipo toolbar.
  - En la página de otras ofertas, si debe aparecer una oferta de tipo cupón, ésta aparecerá en primer lugar.

- Durante el proceso de descarga del software final, el usuario deberá poder visualizar el porcentaje del proceso de descarga.
- Al finalizar la descarga del software final, el instalador del mismo se ejecutará inmediatamente para facilitar su instalación al usuario.
- Se permite que el instalador abra una página web al finalizar.
- El instalador mandará estadísticas a un servidor web informando de todas las acciones que está llevando a cabo.

### **3.2. ANÁLISIS**

El objetivo de este análisis es determinar con exactitud cuáles son las características funcionales para que el generador de gestores de instalación y los instaladores que genera se comporten de acuerdo a los requisitos definidos anteriormente.

Analizando las distintas herramientas y lenguajes existentes para la creación de gestores de instalación, se determinó que lo adecuado para este caso sería utilizar NSIS (Nullsoft Scriptable Install System). Este lenguaje se combinaría con Smarty para poder utilizar plantillas y facilitar el diseño de las diferentes páginas.

Como se desea que el generador pueda estar accesible a través de una web, se creará un documento PHP que recibirá todos los parámetros necesarios para que mediante el uso de las plantillas de Smarty y el código NSIS se pueda generar un instalador de acuerdo a las especificaciones.

### **3.2.1. LENGUAJE NSIS**

Se decidió que el gestor de instaladores debería estar programado en un lenguaje flexible y distribuible a través de internet. Debería ser capaz de generar un instalador de pequeño tamaño para que pudiera descargarse desde cualquier tipo de conexión. Además debía ser flexible para recibir parámetros y poder personalizar cada tipo de instalador en función al programa a descargar, ofertas y características deseadas.

Se escogió NSIS (Nullsoft Scriptable Install System) como lenguaje del instalador ya que cumplía en gran parte todos los requisitos que se exigían en el lenguaje. NSIS se trata de un manejador de scripts Windows que permite realizar simples instaladores que muestran al usuario una sucesión de páginas con diferentes opciones así como otras tareas como generar claves en el registro o generar accesos directos a nuestra aplicación.

NSIS compila estos scripts dentro de un fichero ejecutable, de forma que nuestra aplicación pueda ser distribuida fácilmente. NSIS únicamente añade 34Kb de código de instalador (en la configuración por defecto). Esto unido a que los ficheros de la aplicación son comprimidos según el algoritmo que indique el script o bien el compilador, hace posible que podamos crear instaladores de reducido tamaño.

Para crear un instalador primero debemos escribir un script NSIS. Un script NSIS no es más que un fichero de texto plano con una sintaxis especial. Los scripts NSIS son scripts por lotes en los que cada línea es tratada como un comando. La extensión por defecto de los scripts es .nsi o .nsh.

Un script NSIS puede contener atributos del instalador, páginas, secciones y funciones.

## EJEMPLO

```

;-----
;Include Modern UI
!include "MUI2.nsh"
;-----
;General
;Name and file
Name "Example"
OutFile "Example.exe"

;Default installation folder
InstallDir "$LOCALAPPDATA/Example"
;-----
;Var
var name_example

;-----
;Pages
!insertmacro MUI_PAGE_LICENSE "$\{NSISDIR}\Example\License.txt"
!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_DIRECTORY

;-----
;Languages
!insertmacro MUI_LANGUAGE "English"

;-----
;Installer Sections
Section "Dummy Section" SecDummy

SetOutPath "$INSTDIR"
StrCpy $name_example "Example 1"

;Create uninstaller
WriteUninstaller "$INSTDIR/Example/Uninstall.exe"

SectionEnd

;-----
;Uninstaller Section
Section "Uninstall"

Delete "$INSTDIR/Example/Uninstall.exe"
RMDir "$INSTDIR"

SectionEnd

```

Inclusión del UI moderno

Nombre y fichero de la aplicación

Directorio de instalación

Declaración de variables

Declaración de las macros de las páginas de licencia, bienvenida y directorios

Declaración del idioma

Sección instalación

Sección desinstalación

## ATRIBUTOS DEL INSTALADOR

Los atributos del instalador determinan el comportamiento y la apariencia de nuestro instalador. Con estos atributos podemos cambiar los mensajes que se mostrarán durante la instalación.

Por ejemplo *Name* es el atributo correspondiente al nombre de nuestra aplicación, *InstallDir* será el directorio elegido para instalar la aplicación etc.

## PÁGINAS

Un instalador puede mostrar diferentes páginas al usuario, como por ejemplo la página bienvenida, la de aceptación de licencia, la de selección del directorio de instalación, etc:

```
Page license
Page components
Page directory
Page instfiles
```

Figura 2. Declaración de páginas

En caso de usar el UI moderno al incluir: *!include "MUI.nsh"* usaremos sus macros:

```
!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_LICENSE "licencia.txt"
!insertmacro MUI_PAGE_DIRECTORY
!insertmacro MUI_PAGE_FINISH
```

Figura 3. Declaración de páginas con el UI Moderno

## SECCIONES

En un instalador pueden hacerse categorías de instalación. Y así separar la instalación en varios componentes, dando a elegir al usuario cuales instalar y cuáles no.

```
Section "My Program"
  Setout Path $INSTDIR
  File "My Program.exe"
SectionEnd
```

Figura 4. Declaración de sección

Dentro de cada sección usamos instrucciones que son ejecutadas en tiempo de ejecución. Estas instrucciones, leen y escriben en el registro, crean, borran y copian ficheros y directorios, crean accesos directos etc.

Los desinstaladores también pueden tener varias secciones teniendo como prefijo "un.":

```
Section "Installer Section"
  ...
SectionEnd

Section "un.UninstallerSection"
  ...
SectionEnd
```

Figura 5. Declaración de sección de desinstalación

## FUNCIONES

Las funciones contienen código semejante a las secciones, pero se diferencian de éstas en el modo en que se llaman.

Hay dos tipos de funciones, las definidas por el usuario, que se llaman con la instrucción *Call* y las que se activan cuando ocurren determinados eventos en la instalación:

```
Function function1
  MessageBox MB_OK "Esta es la funcion 1"
FunctionEnd

Function .onInit
  MessageBox MB_YESNO "Esto instalará mi programa ¿Quiere
                        continuar?" IDYES gogogo
  Abort
gogogo:
  call function1
FunctionEnd
```

Figura 6. Declaración de funciones

*Abort* es una función especial que hace que el instalador termine inmediatamente.

### VARIABLES

En este lenguaje se declaran las variables mediante *Var* y se usan utilizando el símbolo \$ delante del nombre:

```
Var BLA ;Declaramos la variable

Section bla
  StrCpy $BLA "123" ;Ahora la variable BLA vale "123"
SectionEnd
```

Figura 7. Declaración y uso de variables

## **3.2.2. USO DE TEMPLATES CON SMARTY**

Smarty es un motor de plantillas para diferentes lenguajes. Más específicamente, esta herramienta facilita la manera de separar la aplicación lógica y el contenido en la presentación.

La principal razón para usar plantillas es que ahorramos tiempo cuando depuramos la aplicación. Una mala práctica de desarrollo consiste en embeber código de programación junto con el diseño y la maquetación. La mayoría de lenguajes de programación permiten esta práctica. Estas son las

funcionalidades que podemos obtener al usar Smarty:

- Plantillas limpias y fáciles de usar por los diseñadores.
- Escalabilidad.
- Mantenimiento más sencillo.
- Depuración óptima del código, al tener ficheros pequeños únicamente con código NSIS.
- Posibilidad de introducir comentarios dentro de las plantillas. Ejemplo: `{* comentario Smarty *}`
- Funciones integradas que facilitan el tratamiento de variables. Ejemplos: `{foreach} {/foreach}`, `{if} {else} {/if}`

El formato de las plantillas es .tpl. Una plantilla puede incluir varias plantillas con lo cual se consigue una estructura de árbol, independiente del script. La sintaxis de Smarty es muy similar a la de PHP, aunque con modificaciones. Por ejemplo, las instrucciones se delimitan con llaves (`{ }`), puede agregarse comentarios dentro de la plantilla (`{ * * }`), existen las estructuras de control IF y FOREACH (para manejo de variables de tipo Array) y otras variaciones de estructuras de control. Las variables se representan con el signo de (`$`).

Para poder incluir código NSIS en las plantillas de Smarty se usan las etiquetas `{literal} {/literal}`. Todo el código que se encuentre entre estas dos etiquetas será tomado literalmente, no siendo interpretado por Smarty.

### EJEMPLO

```
Function Function1
{/literal}
{if $parameter|default:false}
{/literal}
  StrCpy $var "Value"
  ...
{/literal}
{/if}
{/literal}
FunctionEnd
```

Figura 8. Smarty combinado con NSIS

### **3.2.3. PHP Y HTML**

Al tratarse de un instalador que se va a servir de manera web al usuario también se va a usar el lenguaje PHP. Se trata de un lenguaje de programación de código del lado del servidor diseñado para el desarrollo web y el contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

En este proyecto el lenguaje PHP se va a utilizar para poder generar el instalador con todos los parámetros necesarios y para asignar los mismos a las plantillas de Smarty.

Además, como la finalidad es que el generador de gestores de instalación sea accesible desde distintos portales web, se han creado varios ejemplos de sitios de descargas mediante el lenguaje HTML.

### 3.3. DISEÑO

En este apartado se detallará todos pasos llevados a cabo para el diseño del sistema.

En primer lugar, tras el análisis de los lenguajes y tecnologías que se van a utilizar en el desarrollo de este proyecto, se ha diseñado cuál sería el funcionamiento final del mismo.

El siguiente diagrama muestra la secuencia a seguir desde que un usuario solicita la descarga de un programa en un portal de descargas hasta que finalmente lo obtiene.

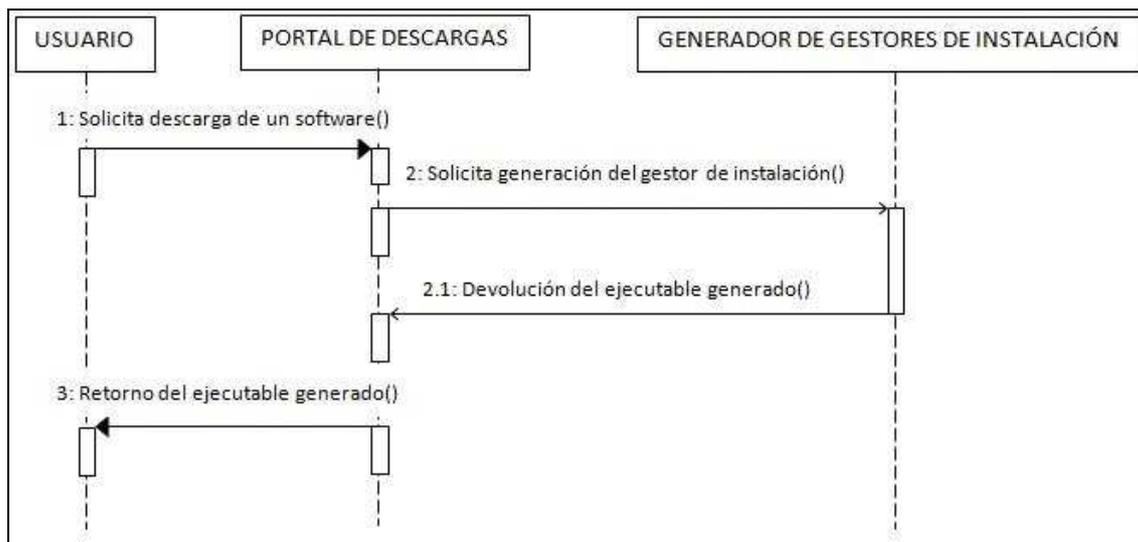


Figura 9. Diagrama de secuencia

Como se puede ver en el diagrama, para el usuario la generación del gestor de instalación es completamente transparente. Es el generador el que se encarga de crear el gestor de instalación con las características y parámetros que tenga establecidos el portal de descargas.

Por otro lado, se pueden dar diferentes casos en los que sea necesario una modificación del generador de gestores de instalación. A continuación, se van a detallar estos casos.

### 3.3.1. AÑADIR NUEVO PORTAL DE DESCARGAS

Si un nuevo portal de descargas quisiera hacer uso del generador de gestores de instalación se tendría que dar de alta el nuevo portal y establecer las características (ofertas y botones que se van a mostrar) que tendrán los gestores de instalación que pertenezcan a ese portal.

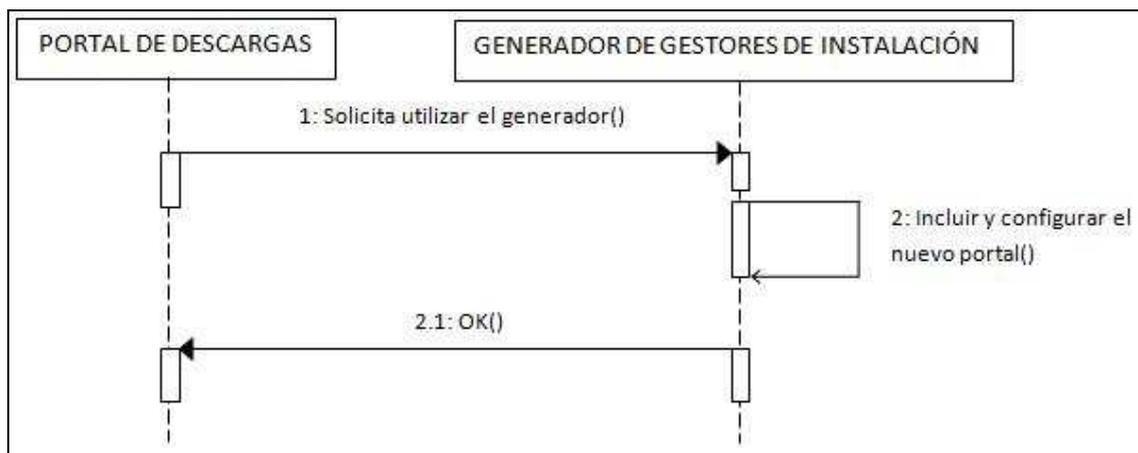


Figura 10. Diagrama de secuencia para añadir un nuevo portal

### 3.3.2. AÑADIR NUEVA OFERTA

También podría darse el caso de que se quisiera añadir una nueva oferta en el generador. En este caso, el proveedor de la oferta debe facilitarnos toda la información de la misma: texto en los diferentes idiomas, imágenes, claves de registro que se crearán, países donde pueden ofrecerse, enlaces a los términos y condiciones y/o política de privacidad, ejecutable de instalación, etc. Con esta información se creará una nueva oferta.

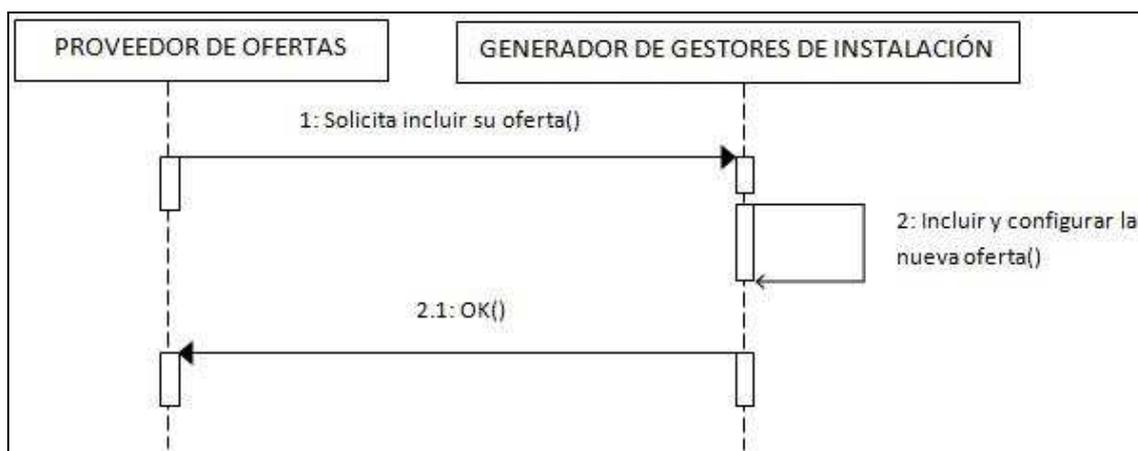


Figura 11. Diagrama de secuencia para añadir una nueva oferta

De esta forma, la oferta pasa a estar disponible para todos los portales de descargas. En el caso de que alguno de ellos quisiera ofrecerla en sus gestores de instalación, tendría que comunicarlo al generador y se añadiría la oferta a la lista de ofertas disponibles para ese portal de descargas.

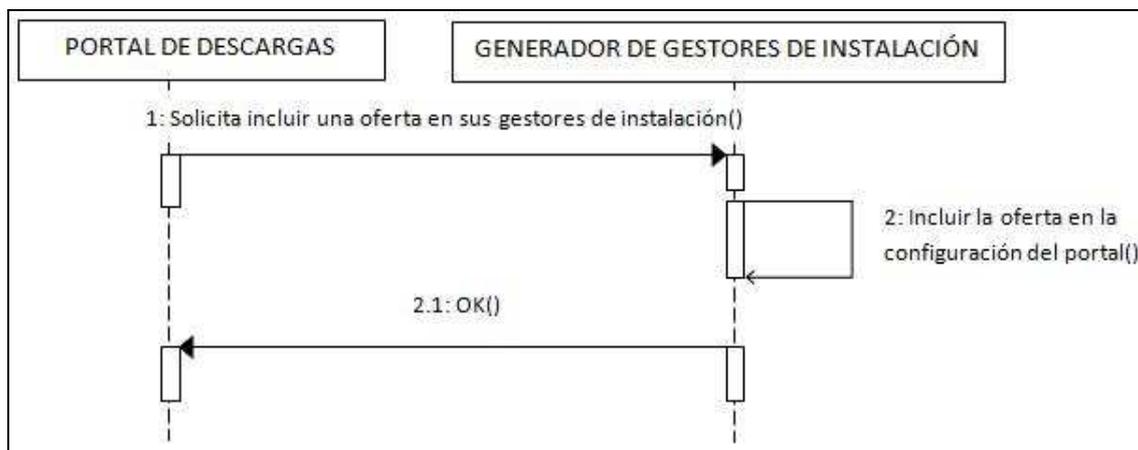


Figura 12. Diagrama de secuencia para añadir una nueva oferta en un portal de descargas

### 3.3.3. AÑADIR NUEVO SOFTWARE

Otro de los casos que se podrían dar, sería el de añadir un nuevo software. En este caso no habría que realizar ninguna tarea en el generador de gestores de instalación. Sería el portal de descargas el que necesitaría la información del nuevo software: nombre, versión, logotipo y dirección web de descarga. Con

esta información, únicamente tendría que pedir la generación del gestor de instalación de este software indicándole al generador la información que le ha facilitado el proveedor del software.

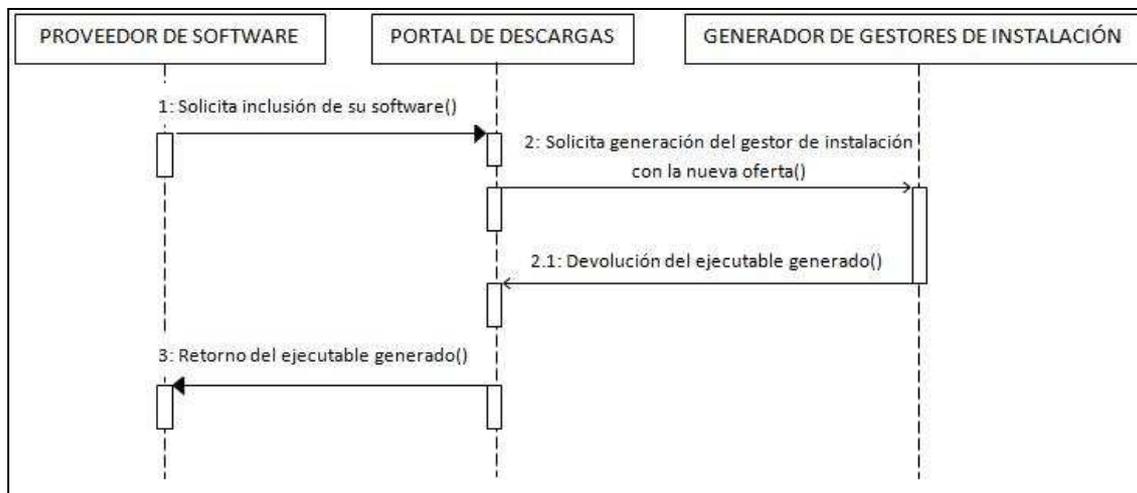


Figura 13. Diagrama de secuencia para añadir un nuevo software

### 3.3.4. DISEÑO DE PANTALLAS Y BOTONES

Con el objetivo de que el gestor de instalación generado sea sencillo y fácil de manejar para el usuario, se han decidido el formato de todas las páginas que se van a poder visualizar en el instalador, la funcionalidad de las mismas y los distintos botones que permitirán al usuario realizar distintas actividades.

En la parte inferior de todas las páginas se mostrarán los distintos botones para poder navegar por el instalador. La aparición de unos botones u otros dependerá del tipo de oferta que se muestre. Así mismo, en todas las páginas excepto la de bienvenida se muestra en la parte superior un logotipo (si no se dispone del logotipo se muestra uno por defecto) y el nombre del software.

Debido a que el instalador puede soportar diferentes idiomas, éste tendrá una apariencia distinta dependiendo de si el idioma es escrito de izquierda a derecha (LTR) o de derecha a izquierda (RTL). Además, el

instalador deberá mostrar caracteres UNICODE correspondientes con la fuente usada para el idioma árabe, chino y japonés.

- PÁGINA DE BIENVENIDA

Esta página mostrará el nombre y logotipo del programa principal y un documento de Licencia.

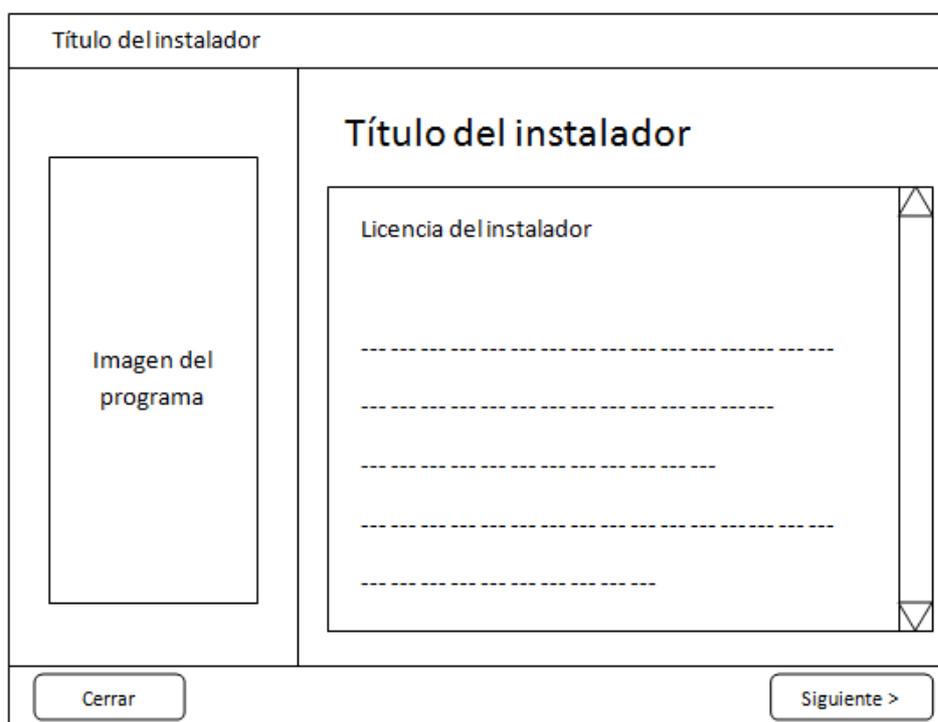


Figura 14: Página de bienvenida

– PÁGINA DE TOOLBAR

Esta página mostrará la oferta de una toolbar y solo aparecerá una única vez en el instalador, lo que quiere decir que el instalador solo ofrecerá cómo máximo una toolbar. La oferta estará compuesta por el título y una imagen de la barra de herramientas y las distintas opciones para su instalación. La mayoría de las toolbar permiten que el usuario elija si quiere instalar la toolbar, cambiar su página de inicio y/o su motor de búsqueda.

Nombre del programa	Logotipo del programa
Título Toolbar	
Imagen Toolbar	
Opciones de instalación de la toolbar	
Rechazar	Aceptar

Figura 15: Página de Toolbar

– PÁGINA/S DE OTRAS OFERTAS

Esta página puede aparecer más de una vez. En ella se puede ver un listado con las distintas ofertas a instalar con un check de selección. Si alguna de las ofertas es de tipo cupón, ésta aparecerá la primera y a continuación todas las demás con un máximo de 4 ofertas por página.

Por cada oferta se mostrará un icono, un texto descriptivo de la misma y un check que permitirá al usuario elegir si quiere instalarla o no. De forma optativa pueden aparecer los enlaces a los términos y condiciones y a la política de privacidad de la oferta.

Nombre del programa	Logotipo del programa
Oferta1 -----	<input checked="" type="checkbox"/>
Oferta2 -----	<input checked="" type="checkbox"/>
Oferta3 -----	<input checked="" type="checkbox"/>
Oferta4 -----	<input checked="" type="checkbox"/>

Cerrar
Siguiente >

Figura 16: Página de otras ofertas

– PÁGINA/S DE OFERTAS A PANTALLA COMPLETA

Esta página puede aparecer más de una vez. En ella se puede ver una información más detallada sobre una oferta en concreto. Para cada oferta tendrá un formato distinto, pero se desea que sea sencilla y atractiva al usuario.

Nombre del programa	Logotipo del programa
<p>Diseño de la oferta de página completa</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p>	
<input type="button" value="Rechazar"/>	<input type="button" value="Aceptar"/>

Figura 17: Página de ofertas a pantalla completa

– PÁGINA DE DESCARGA

En esta página se muestra el progreso de descarga del software final.

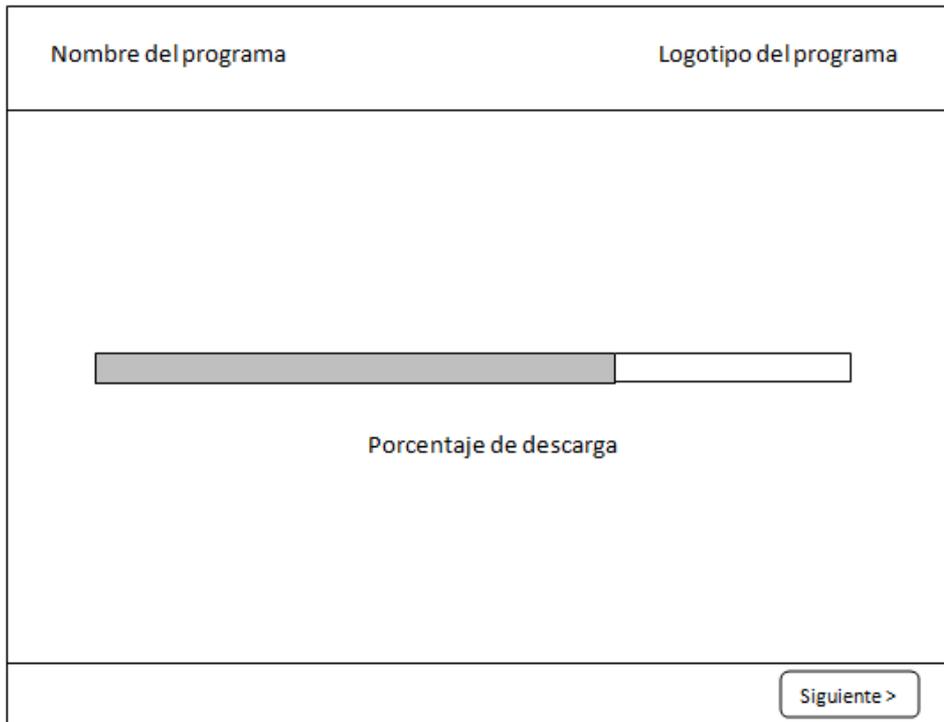


Figura 18: Página de descarga

– PÁGINA FINAL

Esta página mostrará los agradecimientos al usuario por utilizar este instalador para obtener el programa que deseaba.

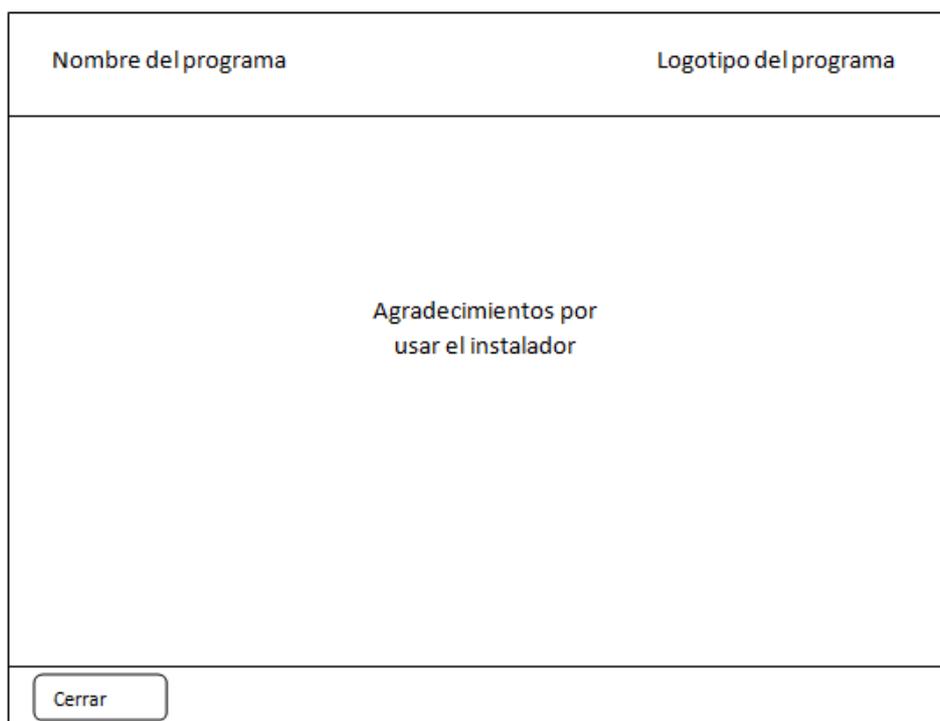


Figura 19: Página final

– BOTON SKIP ALL

Este botón permite al usuario saltarse todas las páginas con ofertas y visualizar directamente la página de descarga.

– BOTON DECLINE

Este botón solamente aparecerá en las páginas de pantalla completa y conllevará la no aceptación de la instalación de la oferta mostrada.

– BOTON ACCEPT

Al igual que el botón Decline, este botón sólo aparecerá en las páginas de pantalla completa y conllevará la aceptación de la instalación de la oferta mostrada.

– BOTON EXIT

Permitirá al usuario salir y cerrar el instalador cuando él desee. Cuando el usuario haga click en este botón se le preguntará si realmente desea cerrar el instalador.

– BOTON NEXT

Este botón pasará a la página siguiente. Si en la página había algún check de aceptación de alguna oferta y estaba marcado se procederá a la instalación de la misma.

– BOTON SKIP THIS

Este botón permitirá pasar a la siguiente página del instalador sin necesidad de aceptar o declinar la instalación de las ofertas que les muestra esa página.



Figura 20: Botones

Por otro lado, el instalador deberá enviar a un servidor web las acciones que va realizando para que puedan ser tratadas y se pueda llevar un control de las instalaciones de ofertas, pensando en una posible optimización del generador.

Las acciones que se van a registrar son:

- Comienzo del instalador.
- Muestra de una oferta.
- Aceptación de una oferta.
- Rechazo de una oferta.
- Salto de una página del instalador.
- Salto de todas las ofertas de páginas de un instalador.
- Comienzo de la descarga del programa.
- Finalización de la instalación de una oferta.
- Muestra de la página final del instalador.

Además de la acción se enviarán distintos parámetros para que el tratamiento de la información sea el más completo posible.

### **3.4. IMPLEMENTACIÓN**

Como se ha explicado en el apartado de Metodología, se ha seguido un modelo de proceso de desarrollo similar al modelo incremental. De esta forma se han ido desarrollando diferentes prototipos de acuerdo con los requisitos.

#### **3.4.1. PRIMERA FASE: INSTALADOR BÁSICO**

En esta primera fase se implementó un gestor de instalación simple con únicamente la página de bienvenida, la de descarga y la de agradecimientos.

Para poder realizar este desarrollo en primer lugar se creó una template de Smarty con las partes básicas del instalador. Esta template recibía como parámetros los diferentes datos parametrizables, como el nombre, la versión, la imagen y la url de descarga del programa a descargar.

Al tratarse de un generador multilingüe se desarrolló un fichero de idioma por cada idioma disponible. Se trata de unos ficheros PHP. Durante la

generación del instalador se carga el fichero del idioma recibido por parámetro. El formato de este fichero es el siguiente:

```

<?php
$h_lang["installer_caption"] = 'Instalador de $program $program_version';
$h_lang["MiscButtonText"]["previous"] = '< Atrás';
$h_lang["MiscButtonText"]["next"] = 'Siguiente >';
$h_lang["MiscButtonText"]["cancel"] = 'Cancelar';
$h_lang["MiscButtonText"]["close"] = 'Cerrar';
$h_lang["MiscButtonText"]["skipall"] = 'Skip All';
$h_lang["MiscButtonText"]["skipthis"] = 'Skip This';
$h_lang["please_wait"] = 'Iniciando Descarga';
$h_lang["init_please_wait"] = 'Inicializando instalación';
$h_lang["lnk_moreinfo"] = '+ info';

$h_lang["MUI_TEXT_WELCOME_INFO_TITLE"] = 'Instalador de $program $program_version';
$h_lang["MUI_TEXT_LICENSE_TITLE"] = '$program $program_version';
$h_lang["MUI_INNERTEXT_LICENSE_TOP"] = 'La descarga comenzará tras la lectura y aceptación del siguiente acuerdo.';
$h_lang["MUI_LICENSEPAGE_BUTTON"] = 'Acepto';
$h_lang["MUI_TEXT_ABORTWARNING"] = '¿Seguro que desea salir de la instalación de $program $program_version?';

$h_lang["page4"]["lbl_ok1"] = 'ha sido descargado exitosamente!';
$h_lang["page4"]["lbl_ok2"] = 'Gracias por usar $partner_desc_web';
$h_lang["page4"]["lbl_ok2_short"] = 'Gracias';
$h_lang["page4"]["lbl_ok3"] = 'Descarga más programas';
$h_lang["page4"]["btn_exit"] = 'Ir a $partner_desc_web';

$h_lang["page4"]["lbl_inf_ok1"] = 'no se encuentra en nuestros servidores premium. ';
$h_lang["page4"]["lbl_inf_ok2"] = 'Clic en el botón para descargarlo desde una url externa';
$h_lang["page4"]["lbl_inf_ok3"] = '';
$h_lang["page4"]["btn_inf_exit"] = 'ir al programa';

$h_lang["accept"] = "Aceptar";
$h_lang["decline"] = "Rechazar";
    
```

Figura 21. Fichero de idioma

Para finalizar con esta fase, se creó un fichero PHP, llamado *gen\_install.php* que se encargaba de recoger todos los parámetros que se reciben (o bien por la línea de comandos o mediante una dirección url), de procesarlos, incluir el fichero de idioma correspondiente, el fichero de licencia y cargar la template de Smarty del instalador. Con estos parámetros y ficheros crea un archivo con extensión .nsh y posteriormente lo compila con el compilador de NSIS para, finalmente, crear un fichero ejecutable con nuestro instalador.

### 3.4.2. SEGUNDA FASE: INSTALADOR CON OFERTAS

Lo más destacable de esta fase es la inclusión de las diferentes ofertas en el instalador.

Para ello, se ha creado una clase PHP llamada *OffersLib.class.inc* donde se definen las ofertas de tipo toolbar, las de tipo cupón y las otras ofertas, así

como los valores y características necesarias para cada una de ellas. Por tanto, se definen tres arrays, uno por cada tipo de ofertas, donde se especifican cuales son las ofertas que corresponden a cada tipo.

```
static function get_array_toolbar_keywords()
{
    return array
    (
        "delta"=>"delta",
        "sweetim"=>"sweetim",
        "incredibar"=>"incredibar",
        "alnaddy"=>"alnaddy",
        "widdit"=>"widdit",
        "gamesbar"=>"gamesbar",
    );
}
```

Figura 22: Array de ofertas tipo toolbar

```
static function get_array_coupon_keywords()
{
    return array
    (
        "coupish" => "coupish",
        "lollipop" => "lollipop",
        "offerbox" => "offerbox",
        "chiliads" => "chiliads",
        "dropinsavings" => "dropinsavings",
        "coupondropdown" => "coupondropdown",
        "ginyas" => "ginyas",
        "strongvault" => "strongvault",
    );
}
```

Figura 23: Array de ofertas tipo cupón

```
static function get_array_otherproducts_keywords()
{
    return array
    (
        "addlyrics" => "addlyrics",
        "optimizerpro" => "optimizerpro",
        "pcspeedup" => "pcspeedup",
        "pagerage" => "pagerage",
        "buzzdock" => "buzzdock",
        "ezlooker" => "ezlooker",
        "dropdowndeals" => "dropdowndeals",
        "freetwittube" => "freetwittube",
        "bestvideodownloader"=> "bestvideodownloader",
        "boxore" => "boxore",
        "m1988" => "m1988",
        "extrafind" => "extrafind",
        "np" => "np",
        "searchresults" => "searchresults",
        "jollywallet" => "jollywallet",
        "playbrite" => "playbrite",
        "qtrax" => "qtrax",
        "browseforthecause" => "browseforthecause",
        "dealply" => "dealply",
        "speedupmypc" => "speedupmypc",
        "driverscanner" => "driverscanner",
        "goplayer" => "goplayer",
    );
}
```

Figura 24: Array de otras ofertas

Además, en esta clase se definen los valores necesarios de cada oferta. Por cada una de ellas se debe implementar una función llamada *set\_vars\_oferta* (donde oferta es el nombre de la misma) en la que se definan los siguientes valores: keyword de la oferta, imágenes o ficheros si son necesarios, url donde se encuentra el fichero ejecutable, parámetros para la ejecución del mismo, países donde se puede ofrecer y todo lo que se va a necesitar en la plantilla de Smarty de esa oferta. Si en el mockup de la ofertas se van a insertar imágenes, éstas deben almacenarse en un directorio y declararlo en la función *set\_vars* de la oferta.

```

static function set_vars_chiliads(&$h_params)
{
    $h_params["offers"]["chiliads"]["offer_keyword"] = "CH";

    copy($h_params["installers_path"]."/images/square/chiliads_tp2.bmp",
        $h_params["installer_src_path"]."/square_chiliads.bmp");
    $h_params["offers"]["images"]["square_chiliads"] = "square_chiliads.bmp";

    if(!isset($h_params["offers"]["chiliads"]["toolurl"]))
    {
        $h_params["offers"]["chiliads"]["toolurl"] = $h_params["media_url"]."/chiliads/ChiliCouponSetup.exe";
    }

    $h_params["offers"]["chiliads"]["base_params"] = " /S ";
    $h_params["offers"]["chiliads"]["base_params_urlencoded"] =
        rawurlencode($h_params["offers"]["chiliads"]["base_params"]);

    if(!isset($h_params["offers"]["chiliads"]["allowed_countries"]))
    {
        $h_params["offers"]["chiliads"]["allowed_countries"] = array(
            "AT"=>array("COUNTRY_DESC"=>"Austria"),
            "CH"=>array("COUNTRY_DESC"=>"Switzerland"),
            "DE"=>array("COUNTRY_DESC"=>"Germany"),
            "GB"=>array("COUNTRY_DESC"=>"United Kingdom"),
            "IE"=>array("COUNTRY_DESC"=>"Ireland"),
        );
    }
}

```

Figura 25: Ejemplo de función *set\_vars*

Por otro lado, se ha creado otro fichero PHP llamado *offers.inc*, el cual debe incluirlo el fichero *gen\_install.php* para establecer las variables de las ofertas que sean necesarias. Además, este fichero establece el orden de aparición de las ofertas en el instalador. El formato de este fichero es el siguiente:

```

<?php
if(OffersLib::check_offer_exists($h_params,"delta"))           { OffersLib::set_vars_delta($h_params); }
if(OffersLib::check_offer_exists($h_params,"incredibar"))     { OffersLib::set_vars_incredibar($h_params); }
if(OffersLib::check_offer_exists($h_params,"sweetim"))        { OffersLib::set_vars_sweetim($h_params); }
if(OffersLib::check_offer_exists($h_params,"coupish"))        { OffersLib::set_vars_coupish($h_params); }
if(OffersLib::check_offer_exists($h_params,"ginyas"))          { OffersLib::set_vars_ginyas($h_params); }

$h_params["toolbar_order_str"] = "delta,gamesbar,widit";
$h_params["coupon_order_str"] = "lollipop,offerbox,strongvault,coupish";
$h_params["otherproduct_order_str"] = "freetwitube,pcspeedup,driverscanner,playbrite,
dealply,boxore,speedupmypc,addlyrics";
$h_params["secondoffers_order_str"] = "lollipop,offerbox,freetwitube,pcspeedup,boxore,
addlyrics,speedupmypc,dealply,playbrite,strongvault,driverscanner,coupish";
    
```

Figura 26: Ejemplo fichero offers.inc

Por otro lado, por cada oferta se debe crear una plantilla de Smarty (extensión .tpl). Estas plantillas están divididas en varias secciones (`{capture}`) que permiten la declaración de variables, funciones, secciones, etc, para posteriormente ser utilizadas en el código del instalador cuando sea necesario, únicamente llamando a esa captura.

Cada oferta tiene definidas las siguientes secciones:

- variables

```
{capture name="offer_variables"} ... {/capture}
```

En esta sección se declaran las variables que se van a usar en la plantilla de la oferta.

- mockup\_normal

```
{capture name="offer_mockup_normal"} ... {/capture}
```

En esta sección se define el mockup de una oferta de tipo toolbar, el cual se mostrará, si cumple con los requisitos para mostrarse, en la primera página del instalador.

- mockup\_secondpage

```
{capture name="offer_mockup_secondpage"} ... {/capture}
```

Esta sección se define en las ofertas de tipo cupón u otros productos. Se trata del mockup que se mostrará en la página de segundas ofertas, si la oferta cumple los requisitos de disponibilidad en el instalador.

- `pageleave`

```
{capture name="offer_pageleave"} ... {/capture}
```

Esta sección se define en todas las ofertas. Se trata del código por el que va a pasar el instalador cuando, después de mostrar la oferta, se pasa de página en el instalador.

- `install`

```
{capture name="offer_install"} ... {/capture}
```

Esta sección define las acciones necesarias para la instalación de la oferta si ésta ha sido aceptada por el usuario.

- `get_availability`

```
{capture name="offer_get_availability"} ... {/capture}
```

En esta sección se realizan las acciones necesarias para determinar si una oferta va a estar disponible para el usuario. Se comprueba que no la tenga ya instalada y que en el país del usuario se esté ofreciendo.

- `functions`

```
{capture name="offer_functions"} ... {/capture}
```

Se definen las funciones que son necesarias para el correcto funcionamiento de la oferta.

- `moreinfo`

```
{capture name="offer_moreinfo"} ... {/capture}
```

Algunas de las ofertas que salen en la segunda página del instalador tienen disponible un enlace para dar más información sobre la misma. En este apartado se definen las acciones necesarias para permitir esta funcionalidad.

- `offer2_pageleave`

```
{capture name="offer_offer2_pageleave"} ... {/capture}
```

Esta sección se define en todas las ofertas de página completa. Al igual que la sección *pageleave*, se trata del código por el que va a pasar el instalador cuando, después de mostrar la oferta, se pasa de página en el instalador.

- popup\_reminder

```
{capture name="offer_popup_reminder"} ... {/capture}
```

Se definen las acciones necesarias para mostrar un popup al usuario cuando éste rechaza una oferta de tipo toolbar que tenga definida esta opción.

Estas secciones son utilizadas por el código del instalador cuando es necesario. Para ello hay que utilizar la instrucción `$smarty.capture` con el nombre de la sección que queremos utilizar.

Finalmente, para permitir que las ofertas se vayan instalando en cuanto se hayan aceptado y no al final de la ejecución del instalador, se ha creado un ejecutable que se ejecutará por cada oferta aceptada una vez que se haya terminado de visualizar la página donde aparece. Este ejecutable también se ha creado con NSIS pero sin plantillas de Smarty. Recibe como parámetros los valores necesarios para la correcta instalación de las ofertas. Estos parámetros son:

- SENDREPORTS: define si se van a enviar las acciones realizadas a un servidor web.
- REPORTURL: define la url del servidor web al que se van a enviar las acciones.
- PARTNERID: identificador del partner (sitio web) que ofrece el instalador.
- OFFERID: identificador de la oferta por partner (sitio web que ofrece el instalador) para poder ser identificado en el servidor web que recibe las acciones.
- OFFERKEYWORD: keyword de la oferta.
- OFFERURL: url donde se encuentra el ejecutable de la oferta.

- OFFERBASENAME: nombre de la oferta.
- OFFERPARAMS: parámetros que necesita el ejecutable de la oferta para ejecutarse.
- CONVERSIONURL: url que se abrirá cuando haya finalizado la instalación de la oferta.

Este ejecutable tiene las instrucciones para instalar todas las ofertas definidas por el instalador. En función al OFFERKEYWORD que recibe realiza las correspondientes a una oferta u otra. A continuación se muestra como ejemplo la función de instalación de una oferta en la que se instala la misma y posteriormente se comprueba si la instalación ha sido correcta comprobando la clave de registro que crea la oferta durante la instalación.

```
Function install_lollipop
    ${exec_offer} $offerurl "$TEMP\LollipopInstaller.exe" $offerparams $0
    StrCpy $return_param $0

    ${If} ${RunningX64}
        ${IfKeyExists} $has_offer 'HKCU' "SOFTWARE" "lollipop"
    ${Else}
        ${IfKeyExists} $has_offer 'HKCU' "SOFTWARE" "lollipop"
    ${EndIf}

    ${If} $has_offer > 0
        StrCpy $return_param "1"
    ${ElseIf} $return_param == "0"
        StrCpy $return_param "-1"
    ${EndIf}

    ${If} $sendreports == "1"
        ToolkitOffers::send_async_url "$reporturl?${vPartnerId}=${partnerid}&${vOfferId}=${offerid}&
        ${vaction}=OfferInstallCompleted&${vparameter}=${return_param}&${vparameter2}=${offerparamsurlencoded}&
        ${vprogram}=${programurlencoded}&${vorigen}=${origen}&${vprod}=LP&${vch}=${channel_urlencoded}"
    ${EndIf}

    ${If} $0 == 0
        ${If} $conversionurl != ""
            ExecShell open "$conversionurl"
        ${EndIf}
    ${EndIf}

    Delete $TEMP\LollipopInstaller.exe
FunctionEnd
```

Figura 27: Ejemplo función install

No es necesario que se compile el código del instalador de ofertas cada vez que se genera un nuevo instalador, ya que este código no cambia. Solo es necesario compilarlo cada vez que se realiza algún cambio en el mismo.

Además, se decidió cuáles serían los parámetros que se enviarían junto con las acciones que se reportaran. Estos parámetros se envían codificados. Los parámetros de cada acción son los siguientes:

- Comienzo del instalador (startedInstall): identificador del partner al que pertenece, identificador de la oferta principal, programa base del instalador e identificadores de todas las ofertas que tiene el gestor de instalación.
- Muestra de una oferta (OfferShown): identificador del partner al que pertenece, identificador de la oferta y programa base del instalador.
- Aceptación de una oferta (OfferAccepted): identificador del partner al que pertenece, identificador de la oferta, programa base del instalador y parámetros que debe recibir el ejecutable de la oferta.
- Rechazo de una oferta (OfferRejected): identificador del partner al que pertenece, identificador de la oferta y programa base del instalador.
- Salto de una página del instalador (SkipThis): identificador del partner al que pertenece, identificador de la oferta saltada y programa base del instalador.
- Salto de todas las ofertas de páginas de un instalador (SkipAll): identificador del partner al que pertenece, identificador de la oferta, programa base del instalador e identificadores de las ofertas saltadas.

- Comienzo de la descarga del programa (StartDownload): identificador del partner al que pertenece, identificador de la oferta principal, programa base del instalador e identificadores de todas las ofertas que tiene el gestor de instalación.
- Finalización de la instalación de una oferta (OfferInstallCompleted): identificador del partner al que pertenece, identificador de la oferta, programa base del instalador, parámetro de salida de la ejecución del ejecutable de la oferta y parámetros que ha recibido el ejecutable de la oferta.
- Muestra de la página final del instalador (EndInstall): identificador del partner al que pertenece, identificador de la oferta principal, programa base del instalador e identificadores de todas las ofertas que tiene el gestor de instalación.

### **3.4.3. TERCERA FASE: FIRMA DEL INSTALADOR**

Durante la implementación del generador de instaladores y a medida que se iban probando los instaladores generados, pude comprobar que es posible que algunos antivirus consideren al instalador un elemento malicioso. Para intentar solventar este inconveniente decidí firmar el instalador para que los usuarios pudieran confiar plenamente en el software que se les ofrece.

Para firmar el ejecutable se ha creado un fichero con extensión PFX con la herramienta *signtool* de Windows. Se trata de un archivo de seguridad encriptado que almacena certificados de seguridad para autenticar a personas o dispositivos.

Debido a que el desarrollo del software se ha llevado a cabo sobre una plataforma Linux, no era posible utilizar la herramienta *signtool* de Windows. Por este motivo se ha usado la herramienta para firma de código OpenSSL.

Se ha creado un pequeño programa en C que recibe como parámetros el nombre del programa que contiene el instalador, la ruta donde se encuentra el instalador generado y la ruta donde queremos que se almacene el nuevo instalador firmado. Este programa utiliza el fichero PFX creado para firmar el ejecutable que le decimos por parámetro.

Para permitir que todos los ejecutables se firmen en tiempo de generación se ha incluido la llamada al programa en C en el fichero *gen\_install.php* al final de la generación del instalador.

De igual forma el ejecutable que instala las distintas ofertas también debe ser firmado. Para ello se ha creado un script en Shell que compila y firma al mismo tiempo el código cuando se realiza algún cambio, de forma que no hay que firmarlo cada vez que se genere un nuevo instalador.

#### **3.4.4. CUARTA FASE: DESARROLLO WEB**

La última fase de la implementación de este proyecto ha sido la creación de cuatro portales de descargas que ofrecen al usuario distintos instaladores.

Cada portal de descargas (partner) ofrecerá diferentes programas mediante el generador de instaladores creado. Todos los instaladores de un mismo partner estarán en el mismo idioma y dispondrán de las mismas ofertas, las cuales se ofrecerán al usuario o no dependiendo de si cumple los requisitos de las mismas, por ejemplo, que no tenga esa oferta ya instalada.

Para realizar esta fase, por cada programa de los portales de descargas se ha creado un fichero PHP que construye una url con los parámetros del instalador que tiene que procesar nuestro generador. El formato del fichero de descarga es el siguiente:

```

<?php
$country = "ES";
$installer_generator_url = "http://sara.installers/down.php";
$url_download = "http://sara.installers/download/36434-665845-vlc-media-player.exe";
$software = "VLC";
$hostname = rawurlencode("vlc.partner1.installers");
$lang = rawurlencode("es");
$installer_lang_utf8 = "Spanish";

$logourl = rawurlencode("/icoinstall/programs/038-vlc.png");

$ua = "generic";
$array_matches_ua = array();
if(preg_match('/(firefox|chrome|msie|opera)/i', $_SERVER['HTTP_USER_AGENT'], $array_matches_ua))
{
    $ua = strtolower($array_matches_ua[1]);
}

$installer_name = $software . " installer.exe";
$software_ue = rawurlencode($software);
$p_var = rawurlencode($_GET['p']);
$url_download = rawurlencode($url_download);

$installer_generator = $installer_generator_url . "?lang=" . $lang . "&hostname=" . $hostname;
$installer_generator .= "&url_download=" . $url_download . "&PRODUCT_NAME=" . $software_ue . "&country=" . $country;
$installer_generator .= "&ud=on&origen=" . $p_var . "&langutf8=" . $installer_lang_utf8;
$installer_generator .= "&logourl=" . $logourl . "&ua=" . $ua;

$s_data = file_get_contents($installer_generator, false,
    stream_context_create(array('http'=>array('method'=>'GET', 'header'=>'X-Real-IP: '
        . $_SERVER['REMOTE_ADDR'] . "\r\n", 'User-Agent: ' . $_SERVER['HTTP_USER_AGENT']))));

header("Content-disposition: attachment; filename=$installer_name");
header("Content-type: application/octet-stream");
header("Content-Length: " . strlen($s_data));

print $s_data;

```

Figura 28: Ejemplo fichero down.php de Dinar

Además, para personalizar más el instalador generado de cada partner, la imagen de la página de bienvenida tendrá el logotipo del mismo.

### 3.4.5. ESQUEMA DEL FUNCIONAMIENTO FINAL

A continuación se muestra un esquema del funcionamiento final del proyecto. Este esquema especifica los ficheros que se usarán y cuál sería la secuencia a seguir en la generación de un gestor de instalación.

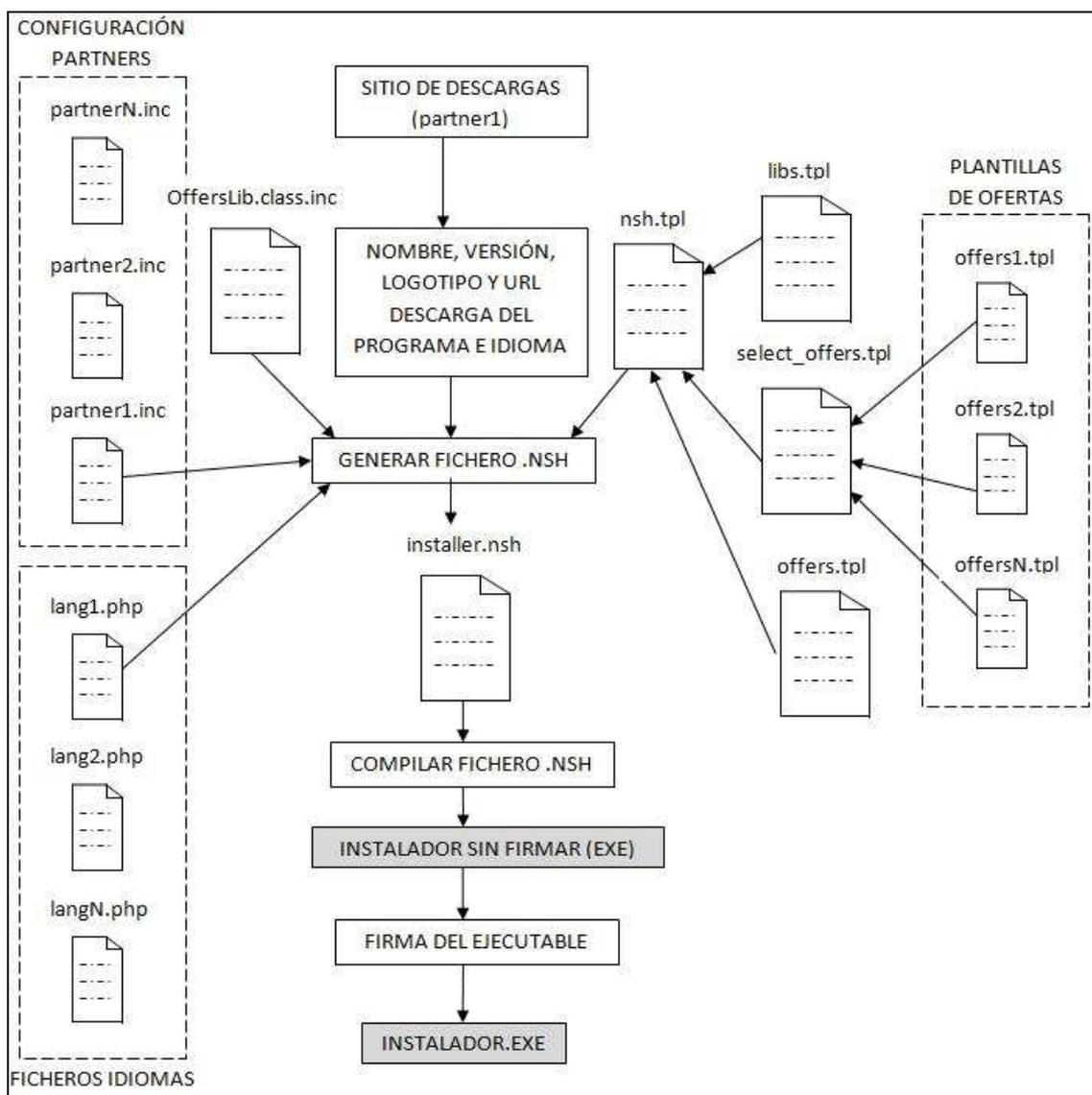


Figura 29. Esquema de generación de un gestor de instalación

Como puede verse en el esquema, el portal de descargas es el encargado de comenzar con la generación del gestor de instalación. El portal indicará al generador los parámetros pertenecientes al software que se va a descargar (nombre, versión, logotipo y url de descarga) y el idioma en el que desea el gestor de instalación.

El generador obtiene todos los parámetros pertenecientes al partner del portal (ofertas que se van a mostrar, botones, documento de licencia, etc) de su fichero de configuración (*partner1.inc*), los textos del idioma indicado (*lang1.php*) y con los parámetros que ha recibido del portal de descargas

comienza con la generación del gestor de instalación.

Para ello, inicializa las variables de los productos pertenecientes al portal de descargas con las funciones `set_vars` de cada una de las ofertas definidas en el fichero `OffersLib.class.inc`. A continuación, genera un fichero `installer.nsh` con las plantillas `nsh.tpl`, `libs.tpl`, `select_offers.tpl` y `offers.tpl`, además de con las plantillas de las ofertas que finalmente vaya a incluir en el gestor de instalación.

El fichero `installer.nsh` tiene todo el código NSIS necesario para crear el gestor de instalación con las características deseadas. A continuación, se compila ese fichero con el compilador de NSIS (`makensis.exe`) y se obtiene un fichero ejecutable.

Finalmente, este fichero ejecutable será firmado para aumentar la confianza del usuario y será devuelto al portal de descargas que lo solicitó.

### **3.4.6. HERRAMIENTAS UTILIZADAS**

Este proyecto se ha implementado combinando diferentes tecnologías. Como se ha detallado anteriormente, se ha utilizado el lenguaje NSIS, el sistema de plantillas de Smarty y los lenguajes PHP y HTML para su uso web.

Se ha utilizado la herramienta *"PhpStorm"*. Se trata de un editor de código rico e inteligente para PHP, HTML y JavaScript con resaltado de sintaxis, configuración de formateo del código extendido, comprobación de errores sobre la marcha y terminación de código inteligente.

El desarrollo del mismo se ha realizado en una única máquina. Para poder realizarlo, se ha desarrollado sobre la plataforma Linux. En ella se han creado diferentes VirtualHosts, que son servidores virtuales. Un servidor virtual

es un directorio que el servidor web sirve con un nombre concreto de máquina.

Para realizar las pruebas sobre una plataforma Windows se ha utilizado el software VirtualBox. Se trata de un programa de virtualización que recrea todos los componentes de un ordenador, desde la CPU hasta la tarjeta gráfica, pasando por la conexión a Internet, tarjeta de sonido y conexiones USB. En esta máquina se ha instalado el software de Windows 7 para poder probar la descarga y la ejecución de los gestores de instalación.

### **3.5. EVALUACIÓN**

En este apartado se va a desarrollar un ejemplo de uso del gestor de instalación y posteriormente se analizarán los resultados obtenidos en algunas pruebas específicas.

#### **3.5.1. EJEMPLO DE FUNCIONAMIENTO**

El generador de gestores de instalación genera estos instaladores en función a los parámetros indicados en el partner perteneciente al sitio de descarga.

El usuario únicamente tiene que seleccionar el botón de descarga del programa en el que esté interesado.



Figura 30. Portal de descargas

Tras la descarga del gestor de instalación, se puede comprobar que el instalador está correctamente firmado, esto permite que haya una mayor confianza en el ejecutable y que los antivirus no lo consideren como objeto malicioso.

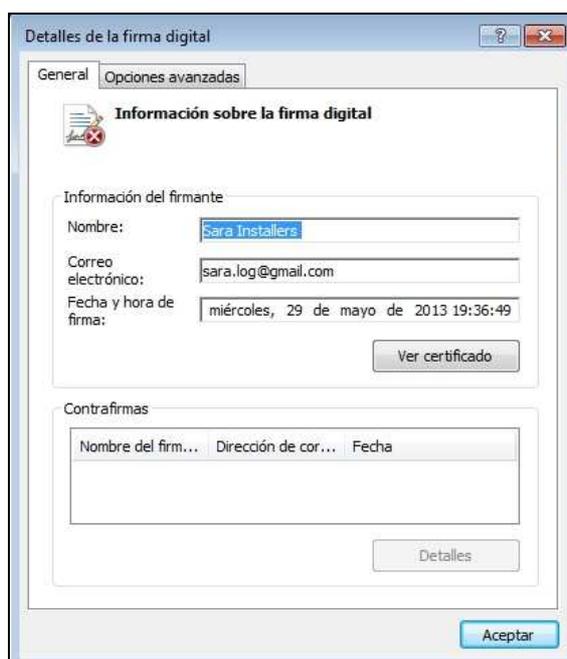


Figura 31. Firma digital

Cuando el usuario ejecuta el gestor de instalación se muestra la página de bienvenida con el logotipo, el nombre del programa que desea descargarse y el contrato de licencia.



Figura 32. Página de Bienvenida

Al mismo tiempo, se envía la acción al servidor que recibe las acciones que se realizan.

```
IP -- [dd/MM/yyyy:HH:mm:ss] "GET  
/?PartnerId=partner2&OfferId=200&action=startedInstall&program=VLC%20&parameter=&orig  
en=&of=200%2C211%2C214%2C218%2C219%2C220%2C221%2C222%2C224%2C226%2C228  
%2C229%2C231%2C232%2C233%2C234%2C235%2C213%2C215%2C216
```

Figura 33. Log de la acción StartedInstall

Al pasar a la siguiente página mediante el botón “Siguiente” se muestra la página del producto de tipo Toolbar. En ocasiones, es posible que esta página no se muestre o bien porque el sitio web no tiene configurado mostrar la página o bien porque no corresponde mostrarla según los requisitos de la oferta (el usuario ya la tiene instalada en su equipo, pertenece a un país donde no se puede instalar la oferta,...)

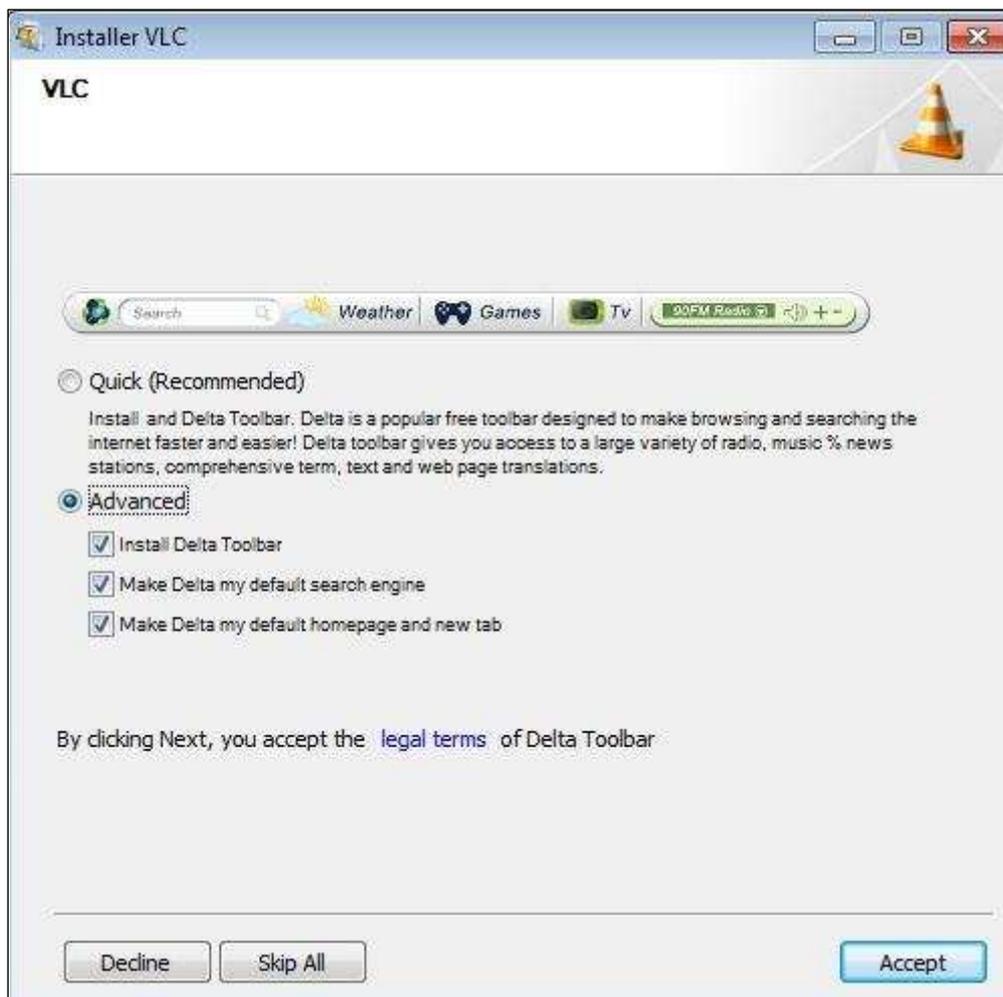


Figura 34. Página de Toolbar

Al mostrar la página también se envía la acción al servidor. En este caso se trata de la acción *OfferShown* que se enviará siempre que se muestre una oferta.

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET  
/?PartnerId=partner2&OfferId=200&action=OfferShown&program=VLC%20
```

Figura 35. Log de la acción OfferShown

Si el usuario acepta la oferta, al pasar la página se comenzará con la instalación de la misma. Se reportará una acción *OfferAccepted* y cuando finalice la instalación de la oferta se reportará la acción *OfferInstallCompleted*.

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET  
/?PartnerId=partner2&OfferId=232&action=OfferAccepted&parameter=%20%2FS%20%2FPART  
NER%3Dvita%20%2FCHANNEL%3D111%20%2FOPTIMIZE&program=VLC%20
```

Figura 36. Log de la acción OfferAccepted

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET  
/?PartnerId=partner2&OfferId=232&action=OfferInstallCompleted&parameter=1&parameter2=%  
20%2FS%20%2FPARTNER%3Dvita%20%2FCHANNEL%3D111%20%2FOPTIMIZE&program=  
VLC%20
```

Figura 37. Log de la acción OfferInstallCompleted

Al mismo tiempo, si el usuario rechaza la oferta, la acción que se enviará al servidor será la de *OfferRejected*.

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET  
/?PartnerId=partner2&OfferId=234&action=OfferRejected&program=VLC%20
```

Figura 38. Log de la acción OfferRejected

La siguiente página que se le mostrará al usuario es la de otras ofertas. En esta página se muestran como máximo 4 ofertas y si alguna de ellas es un producto de tipo cupón, éste aparecerá en primer lugar. El usuario podrá aceptar o rechazar las ofertas mediante los checks que se muestran. Esta página puede aparecer una o dos veces dependiendo de las ofertas que tenga

configurado el sitio de descargas.

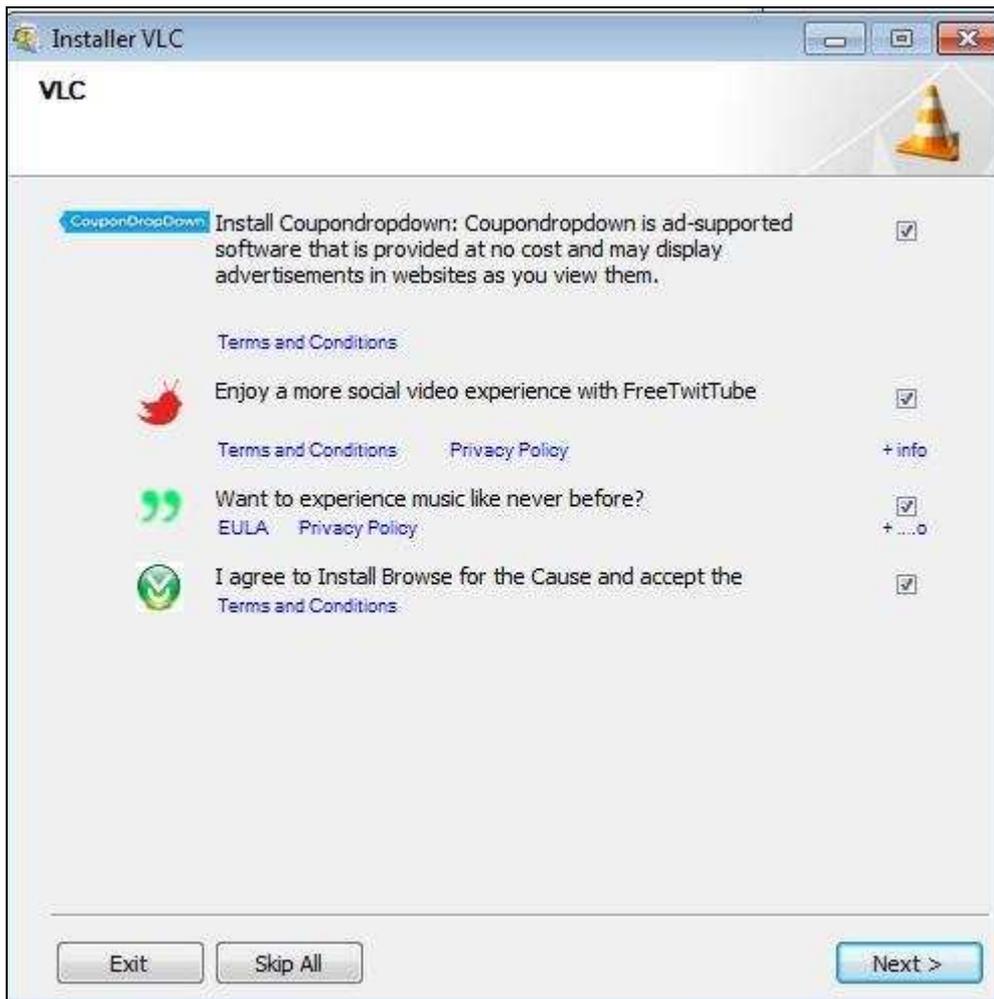


Figura 39. Página de otras ofertas

Al igual que con la oferta de la página de toolbar, en esta página se envían tantas acciones *OfferShown* como ofertas se muestren en la página.

```
IP -- [dd/MM/yyyy:HH:mm:ss]"GET
/?PartnerId=partner2&OfferId=228&action=OfferShown&program=VLC%20
IP -- [dd/MM/yyyy:HH:mm:ss]"GET
/?PartnerId=partner2&OfferId=232&action=OfferShown&program=VLC%20
IP -- [dd/MM/yyyy:HH:mm:ss]"GET
/?PartnerId=partner2&OfferId=233&action=OfferShown&program=VLC%20
IP -- [dd/MM/yyyy:HH:mm:ss]"GET
/?PartnerId=partner2&OfferId=234&action=OfferShown&program=VLC%20
```

Figura 40. Log de la acción OfferShown en una misma página

La siguiente página que se le muestra al usuario al pulsar sobre el botón Siguiente, es la página de oferta a pantalla completa. Al igual que la anterior página, ésta puede aparecer más de una vez dependiendo de las ofertas que se hayan establecido.

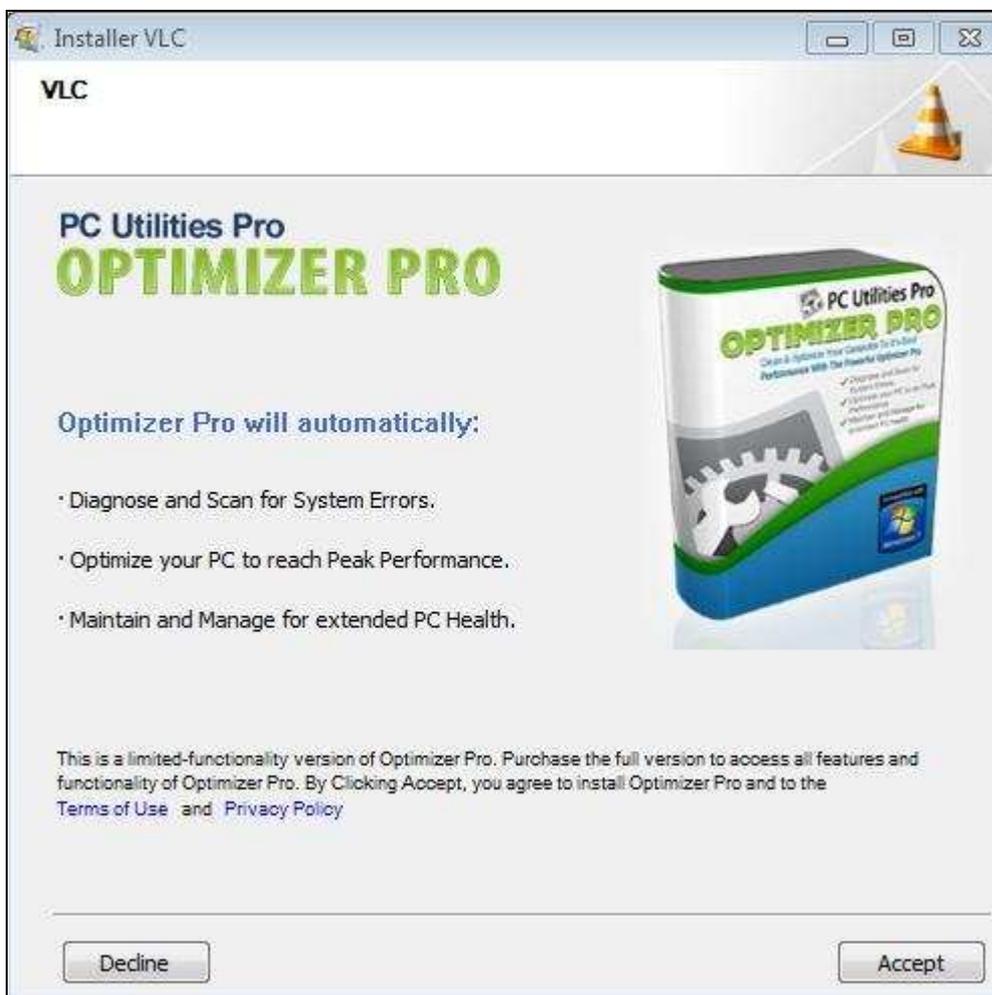


Figura 41. Página de ofertas a pantalla completa

Al igual que con las demás también se reporta la acción *OfferShown*.

Si el usuario acepta la oferta mediante el botón de Aceptar se enviará la acción *OfferAccepted* y si el usuario rechaza la oferta se enviará la acción *OfferRejected*.

La siguiente página que ve el usuario es la página de descarga. En esta página se muestra una barra de progreso de la descarga de software y el

tiempo estimado que queda para que finalice la misma. En esta página el botón de Siguiente no aparece habilitado hasta que finaliza la descarga.

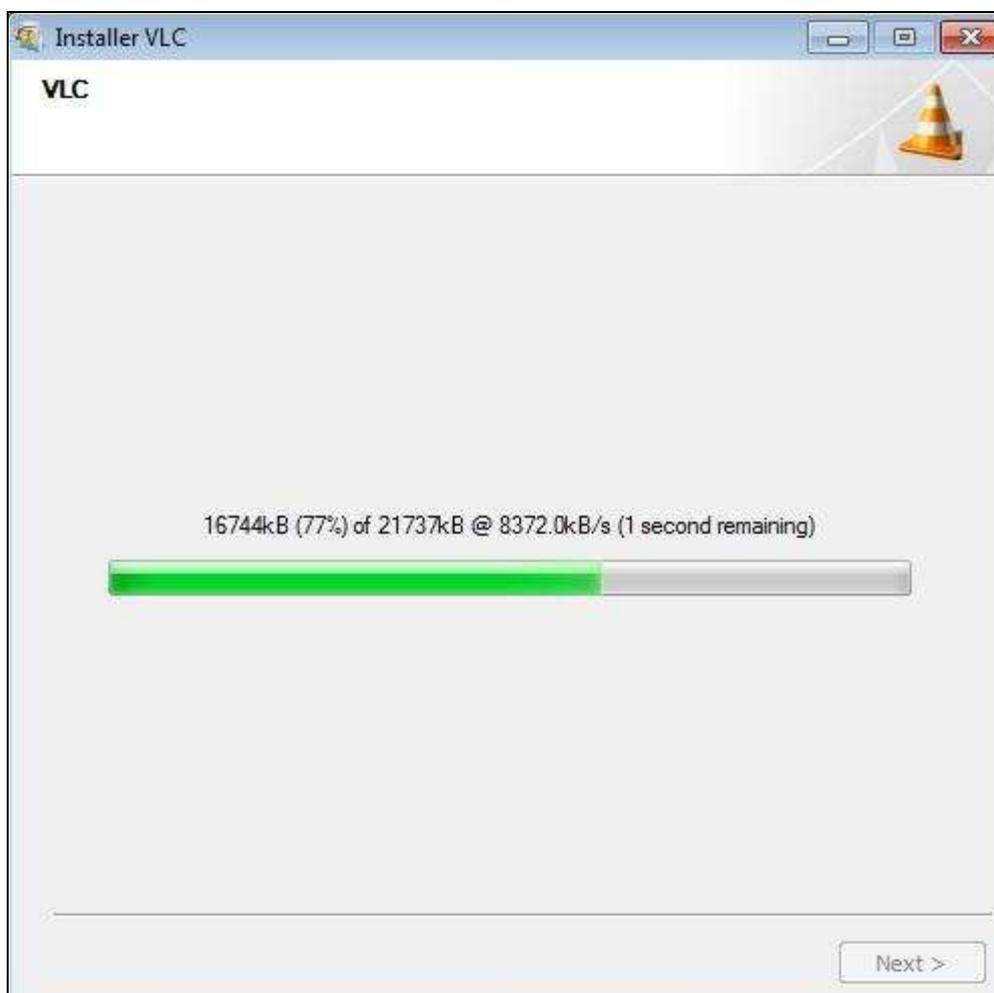


Figura 42. Página de descarga

La acción que se reporta en este caso es la de *StartDownload*.

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET  
/?PartnerId=partner2&OfferId=200&action=StartDownload&program=VLC%20&of=200%2C211  
%2C214%2C218%2C219%2C220%2C221%2C222%2C224%2C226%2C228%2C229%2C231%2  
C232%2C233%2C234%2C235%2C213%2C215%2C216
```

Figura 43. Log de la acción StartDownload

Cuando finaliza la descarga y el usuario pulsa el botón de Siguiente se muestra la página final. En esta página se pueden visualizar los agradecimientos por usar este gestor de instalación y se abre automáticamente la ejecución del programa descargada para que el usuario se lo pueda instalar en su equipo.

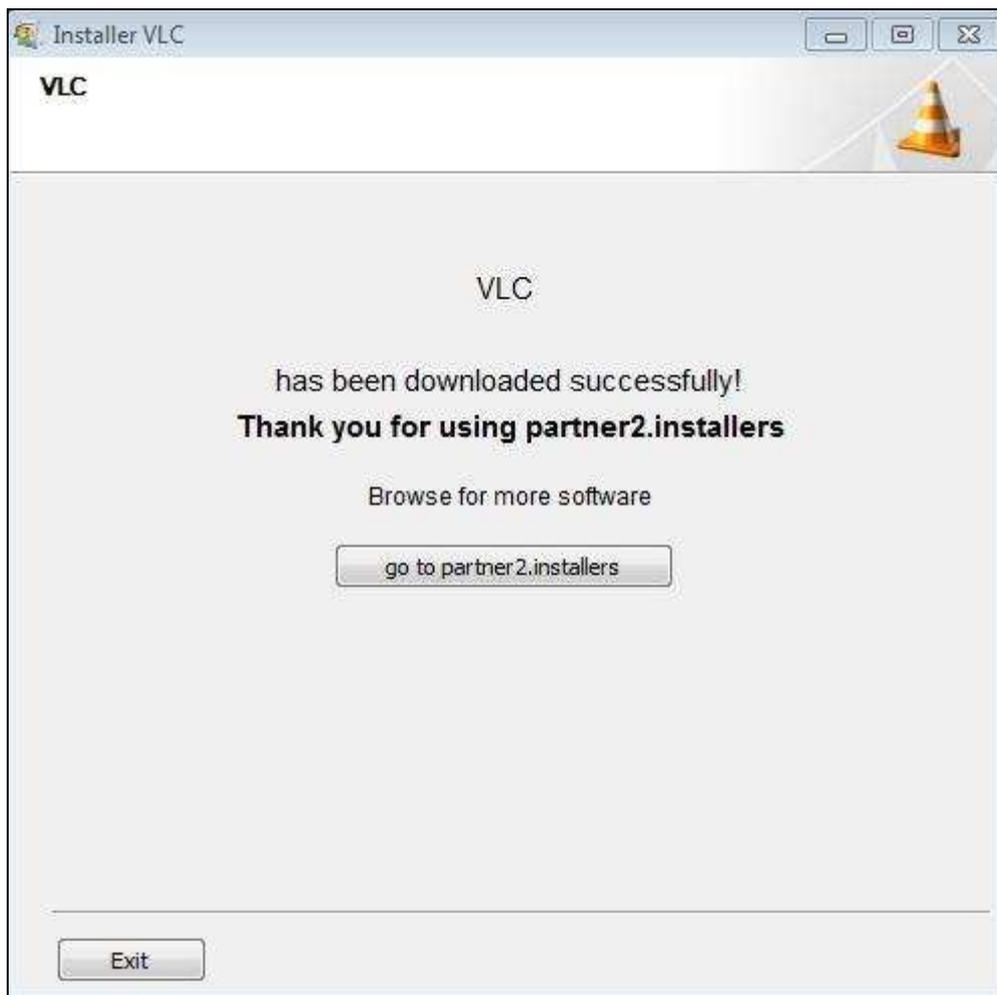


Figura 44. Página final

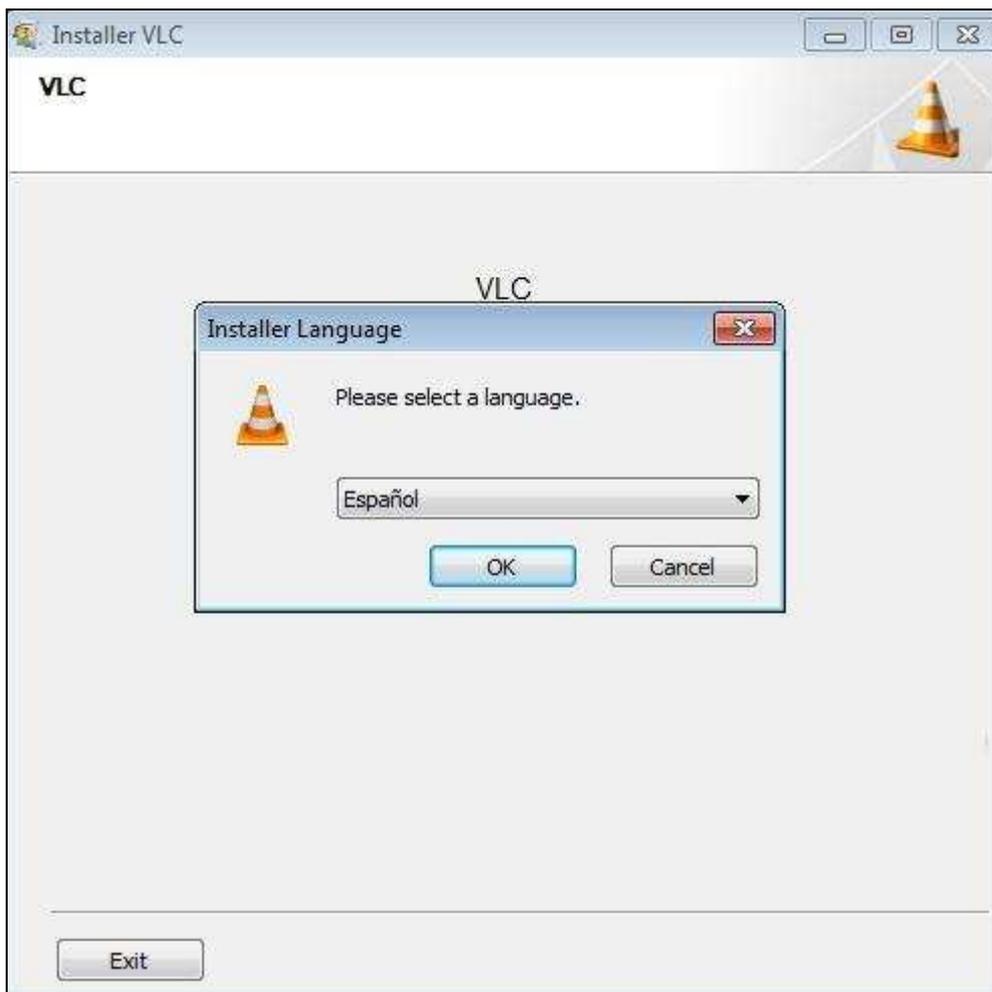


Figura 45. Ejecución del programa

En esta página la acción que se reporta es la de *EndInstall*.

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET  
/?PartnerId=partner2&OfferId=200&action=EndInstall&program=VLC%20&o  
f=200%2C211%2C  
214%2C218%2C219%2C220%2C221%2C222%2C224%2C226%2C228%2C229%2C231%2C232  
%2C233%2C234%2C235%2C213%2C215%2C216
```

Figura 46. Log de la acción EndInstall

En todas las páginas del instalador se muestra el botón de cerrar. Si el usuario desea salir del instalador debe pulsar este botón y se le preguntará si está seguro que quiere salir.

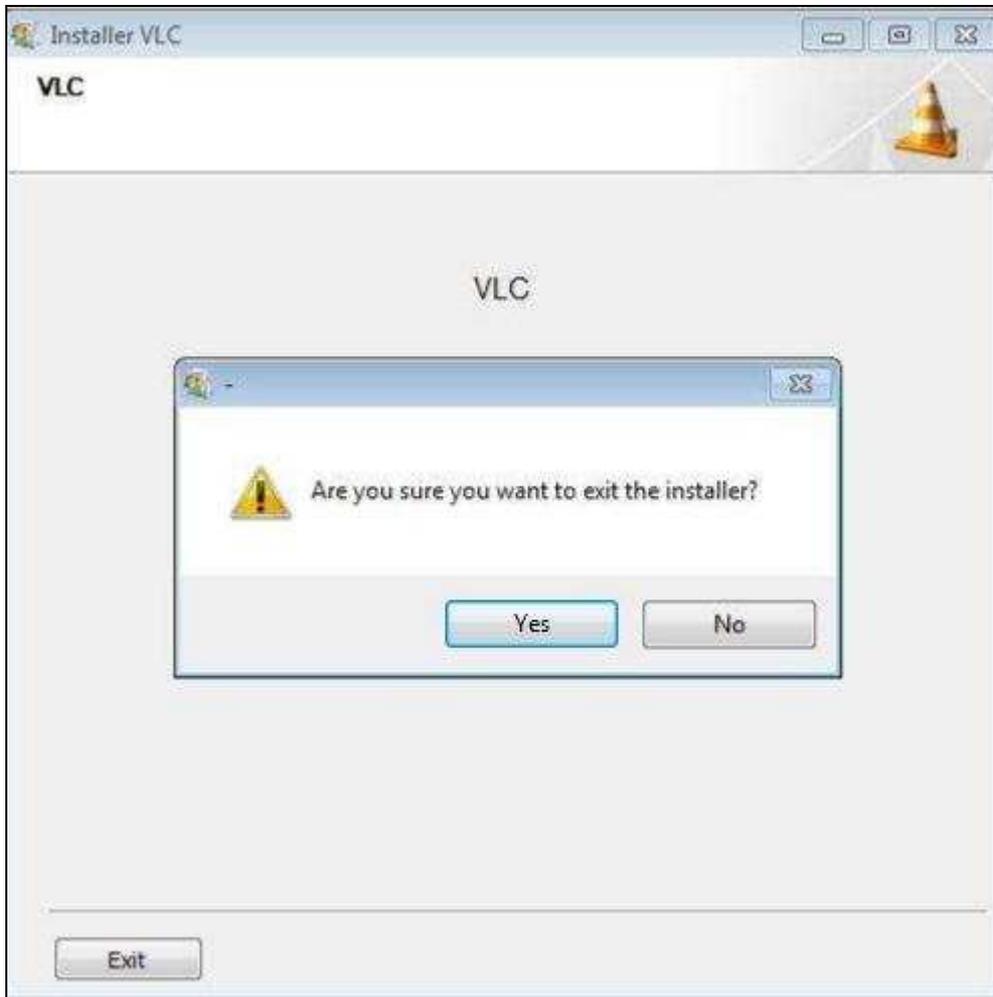


Figura 47. Salir del gestor de instalación

Al mismo tiempo, es posible que los gestores de instalación ofrezcan al usuario la posibilidad de saltar todas las ofertas o una en concreto. En estos casos si el usuario pulsara el botón Skip All la acción que se reportaría sería.

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET
/?PartnerId=partner2&OfferId=200&action=SkipAll&program=VLC%20of=200%2C211%2C214
%2C218%2C219%2C220%2C221%2C222%2C224%2C226%2C229%2C231%2C232%2C233%2
C234%2C235%2C213%2C215%2C216
```

Figura 48. Log de la acción SkipAll

Si el usuario pulsara el botón Skip This, la acción sería:

```
IP - - [dd/MM/yyyy:HH:mm:ss]"GET
/?PartnerId=partner2&OfferId=200&action=SkipThis&program=VLC%20&of=20
```

Figura 49. Log de la acción SkipThis

### 3.5.2. EJEMPLO DE INSTALADOR EN ÁRABE

A continuación se muestran las diferentes páginas de in instalador en árabe.



Figura 50. Página de bienvenida en árabe



Figura 51. Página de toolbar en árabe



Figura 52. Página de otras ofertas en árabe

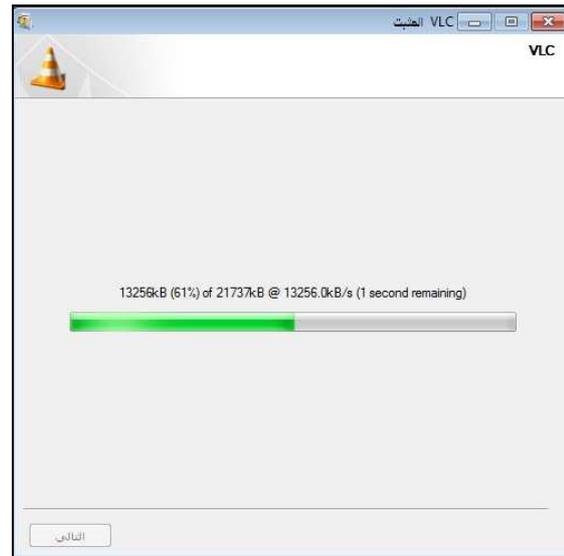


Figura 54. Página de descarga en árabe

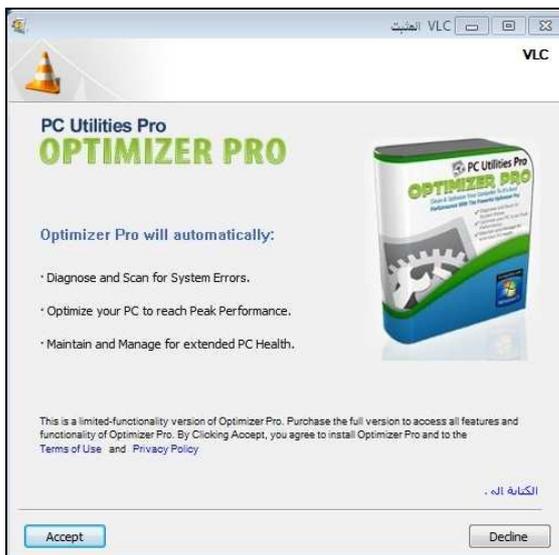


Figura 53. Página de oferta a pantalla completa en árabe

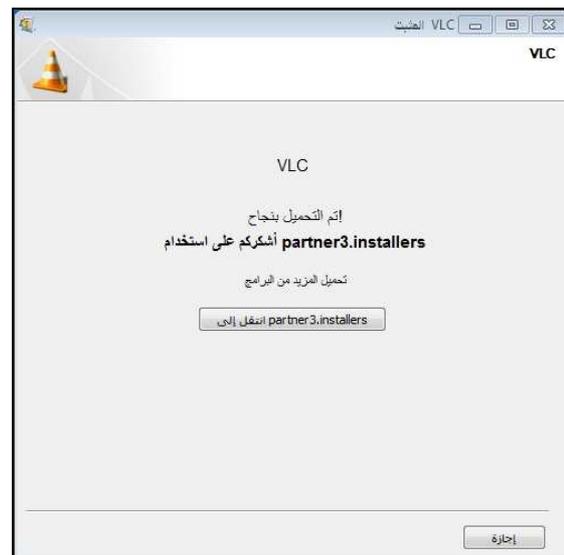


Figura 55. Página final en árabe

### 3.5.3. EJEMPLO DE INSTALADOR EN JAPONÉS

A continuación se muestran las diferentes páginas de in instalador en japonés.

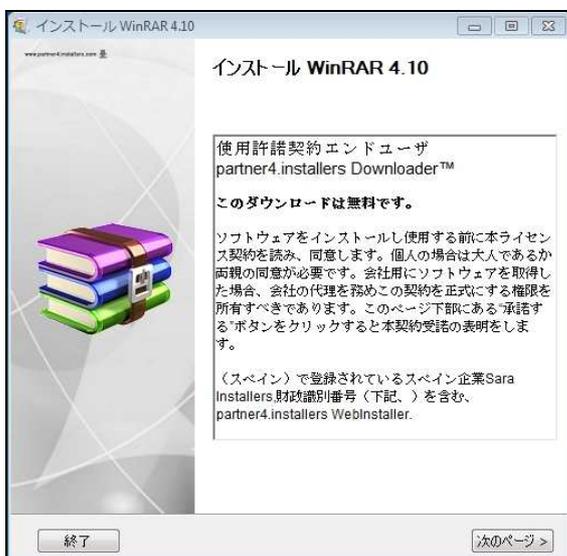


Figura 56. Página de bienvenida en japonés



Figura 58. Página de otras ofertas en japonés



Figura 57. Página de toolbar en japonés

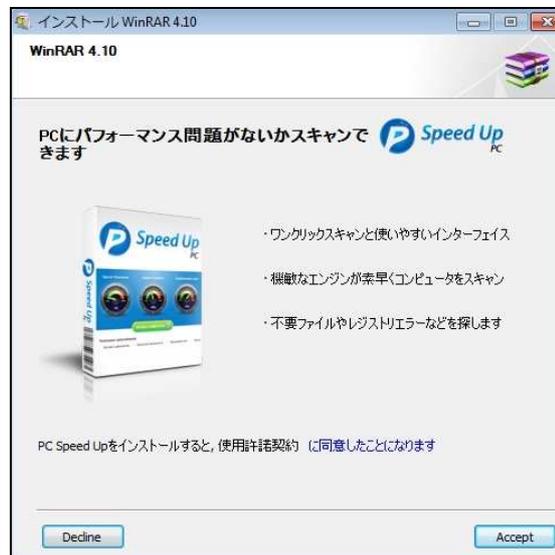


Figura 59. Página de oferta a pantalla completa en japonés

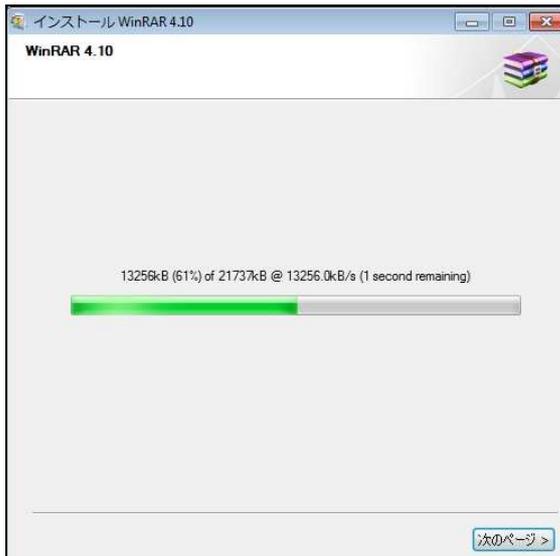


Figura 60. Página de descarga en japonés



Figura 61. Página final en japonés

## **4.CONCLUSIONES Y LÍNEAS FUTURAS**

### **4.1. LOGROS PRINCIPALES ALCANZADOS**

El objetivo de este proyecto era crear un sistema de calidad, robusto y flexible que permitiera generar gestores de instalación a través de un portal web de descargas de forma transparente al usuario. Al mismo tiempo, otro de los objetivos era que el gestor de instalación reportara informes de estadísticas para un posible estudio de viabilidad y mejora del gestor.

Estos objetivos se han cumplido, ya que se han desarrollado varios portales de descargas que permiten que cualquier usuario obtenga el software deseado y éste se le ofrezca mediante un gestor de instalación ofreciéndole al mismo tiempo otros productos para su instalación.

Además, el generador creado puede ser usado por gran variedad de personas, ya que genera instaladores en varios idiomas.

En cuanto a mi experiencia personal comentar que, debido a que era la primera vez que programaba en NSIS, me resultó difícil adecuarme a la programación de instaladores y tuve algunas dificultades en la creación de páginas o eventos de instalación. El uso de las plantillas de Smarty me permitió tener una mayor flexibilidad a la hora de programar las distintas ofertas que incluye el gestor de instalación.

Durante la implementación, he aprendido el lenguaje NSIS y el uso de plantillas Smarty. Además he podido aumentar mis conocimientos en la tecnología PHP.

Por tanto, me ha resultado muy útil la creación del presente proyecto, ya que me ha permitido aumentar mis conocimientos adquiridos durante el estudio de esta ingeniería y al mismo tiempo he adquirido nuevos conocimientos sobre la creación de gestores de instalación.

## 4.2. LÍNEAS FUTURAS

Se exponen a continuación varias propuestas como líneas futuras a este proyecto:

- Recoger las estadísticas enviadas por el gestor de instalación. Para esto se podría crear un servidor donde se enviaran estas estadísticas. Este servidor podría almacenar los datos que recibe en una base de datos, para posteriormente ser analizados y estudiados.
- Definir el orden de las ofertas por país ya que actualmente hay un único orden de ofertas común para todos los países. De esta forma se podrían ofrecer ordenadamente los productos más populares dependiendo del país.
- Añadir más idiomas en el gestor de instalación. Aunque actualmente está implementado para 15 idiomas se podrían añadir más a la lista, como por ejemplo el finlandés, catalán, polaco, ...
- Mostrar un vídeo en la página de descarga. De esta forma este vídeo podría considerarse como otra oferta y permitiría al usuario disfrutar del mismo durante el proceso de descarga, ya que puede demorarse unos minutos.
- Añadir más formatos de ficheros en el gestor de instalación. Actualmente el instalador solo es capaz de comenzar la ejecución de ficheros .exe. Podría abrir ficheros .rar o poner de fondo de pantalla una imagen.
- Crear un “gestor de instalación exprés”. Se trataría de un instalador con una página inicial en la que el usuario podría elegir entre dos opciones. La primera sería si quiere instalar todas las ofertas que se le van a ofrecer automáticamente sin mostrar sus mockups y la segunda opción es continuar con el gestor de instalación de manera normal.

## 5. REFERENCIAS

1. NSIS, [http://nsis.sourceforge.net/Main\\_Page](http://nsis.sourceforge.net/Main_Page) (Último acceso: mayo de 2013).
2. Foro de NSIS, <http://forums.winamp.com/forumdisplay.php?s=&forumid=65> (Último acceso: mayo de 2013).
3. Smarty, <http://www.smarty.net/> (Último acceso: febrero 2013).
4. R. S. Pressman. "Ingeniería del Software. Un enfoque práctico". McGraw-Hill, 1998.

## ANEXO A

### ORGANIZACIÓN DE FICHEROS

A continuación se describe la organización de los ficheros contenidos en el CD.

- \installers: Este directorio contiene todos los archivos necesarios para la ejecución del software. Entre otros podemos encontrar:
  - \sara\_installers: Contiene todos los ficheros pertenecientes al generador de gestores de instalación.
  - \sign\_installers: Se encuentran los ficheros necesarios para la firma del ejecutable.
  - \offers: Contiene los ejecutables de las ofertas.
  - \download: Este directorio contiene los ejecutables de los programas finales del gestor de instalación.
  - \report: Directorio donde se encuentran los ficheros de log que almacenan las acciones que se realizan en los gestores de instalación.
  
- \Ejemplos: En este directorio se encuentra ejemplos de gestores de instalación en diferentes idiomas.
  
- \readme.txt: Fichero que contiene información relativa sobre el proyecto.
  
- \memoria: Directorio donde se encuentra almacenada la memoria en formato PDF del proyecto.