

CONTENIDO

Descripción	2
Prueba No. 1: Creación de un proyecto	2
Prueba No. 2: Creación de un proyecto sin parámetros de entrada	3
Prueba No. 3: Creación de un proyecto con un nombre que ya existe	3
Prueba No. 4: Reemplazar un proyecto existente	3
Prueba No. 5: Creación de un MIDlet.....	4
Prueba No. 6: Creación de un MIDlet teniendo asociado un Form.....	4
Prueba No. 7: Creación de un MIDlet teniendo asociado un Canvas	5
Prueba No. 8: Creación de un MIDlet teniendo asociado un List.....	6
Prueba No. 9: Creación de un MIDlet teniendo asociado un TextBox	7
Prueba No. 10: Creación de una clase para administrar un RecordStore.....	8
Prueba No. 11: Creación de una conexión HTTP.....	9
Prueba No. 12: Adición de un Fólder al proyecto	10
Prueba No. 13: Configuración del WTK a un Proyecto	10
Prueba No. 14: Selección de un Emulador.....	11
Prueba No. 15: Adición de un emulador al WTK 2.2.....	11
Prueba No. 16: Actualizar un Proyecto.....	12
Prueba No. 17: Compilación de un Proyecto.....	12
Prueba No. 18: Ejecución de un Proyecto	13
Prueba No. 19: Compilación de un archivo con errores (falta de un ;).	13
Prueba No. 20: Compilación de un proyecto con errores (referencia desconocida)..	14
Prueba No. 21: Eliminación de un archivo.....	14
Prueba No. 22: Adicionar una biblioteca de clases a PROMETEO.....	15
Prueba No. 23: Adicionar una biblioteca a un Proyecto.....	16
Prueba No. 24: Compilación de un proyecto con una biblioteca asociada	16
Prueba No. 25: Crear un archivo JAR	17
Prueba No. 26: Desvincular una Biblioteca de un Proyecto.....	18

Descripción

El presente documento contiene las respuestas de la plataforma a diferentes pruebas, incluyendo por cada prueba el resultado esperado, el resultado obtenido y la respectiva discusión en donde haya lugar.

Aunque las pruebas se muestran en el sistema operativo Windows, mientras no se diga lo contrario debe asumirse que los resultados esperados y obtenidos se dan de igual forma sobre la plataforma Linux.

Prueba No. 1: Creación de un proyecto

Descripción: se utilizará el wizard Nuevo Proyecto para crear un proyecto en el entorno de desarrollo.

Resultado(s) esperado(s):

- Se creará un directorio con el nombre Proyecto1 en la carpeta C:/prometeo/pruebas. Dentro de la carpeta Proyecto1 estarán los directorios src, res, bin, doc y el archivo Proyecto1.ppr.
- El archivo Proyecto1.ppr debe presentar el siguiente contenido:

```
Proyecto1.ppr
- <proyecto>
  <nombre>Proyecto 1.ppr</nombre>
  <path />
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
  <midlets />
  <paquetes />
</proyecto>
```

- En la GUI principal de la plataforma aparecerá un nodo en el árbol que muestra la estructura del proyecto, con el nombre Proyecto1.ppr.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 2: Creación de un proyecto sin parámetros de entrada

Descripción: se utilizará el wizard Nuevo Proyecto y se intentará crear un proyecto sin pasar en la GUI los parámetros nombre del proyecto y directorio de creación.

Resultado(s) esperado(s):

- Aparecerá un mensaje de error indicando que el campo Nombre no debe estar vacío.
- Si se llena el campo Nombre, y se intenta crear el proyecto, aparecerá un mensaje de error indicando que campo directorio no puede estar vacío.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 3: Creación de un proyecto con un nombre que ya existe

Descripción: se utilizará el wizard Nuevo Proyecto y se intentará crear un proyecto de nombre Proyecto1 en el directorio C:/prometeo/pruebas.

Resultado(s) esperado(s):

- Aparecerá un mensaje de advertencia indicando que el directorio Proyecto1 ya existe en C:/prometeo/Pruebas y preguntando si se desea reemplazar el directorio existente.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 4: Reemplazar un proyecto existente

Descripción: Partiendo de la prueba anterior, en la ventana de advertencia el desarrollador selecciona la opción SI (Reemplazar el proyecto existente).

Resultado(s) esperado(s):

- Se elimina el directorio que ya existe (Proyecto1) con todo su contenido y en su lugar se crea un nuevo proyecto de nombre Proyecto1 igual al creado en la prueba1.
- En la GUI principal, en el árbol del proyecto aparece un nodo con el nombre Proyecto1.ppr.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 5: Creación de un MIDlet.

Descripción: Se utilizará el wizard Nuevo MIDlet para crear un MIDlet de Nombre MIDlet1 en un paquete de nombre paquete1 en el proyecto Proyecto1.

Resultado(s) esperado(s):

- Se crea un directorio de nombre paquete1 en el directorio C:/prometeo/pruebas/Proyecto1/src.
- Dentro del directorio paquete1 se creará el archivo MIDlet1.java
- En la GUI principal de Prometeo se muestra en el editor de código, el código correspondiente al archivo MIDlet1.java.
- En la GUI principal de Prometeo, en el árbol del proyecto se mostrará el nodo correspondiente al paquete paquete1.
- El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path />
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 6: Creación de un MIDlet teniendo asociado un Form.

Descripción: con el wizard Nuevo MIDlet se creará otro MIDlet de nombre MIDlet2, en el paquete paquete1. Se marcará la opción crear Desplegable, al nombre de la clase se le

dará Form1 y al Título "Hola Prometeo", en el tipo del desplegable se escogerá javax.microedition.lcdui.Form.

Resultado(s) esperado(s):

- En el directorio C:/prometeo/pruebas/Proyecto1/src/paquete1 crean los archivos MIDlet2.java y Form1.java.
- En la GUI principal de Prometeo, aparece en el editor de código el código correspondiente al archivo Form1.java.
- El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path />
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
  <paquete1.MIDlet2 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 7: Creación de un MIDlet teniendo asociado un Canvas

Descripción: con el wizard Nuevo MIDlet se creará otro MIDlet de nombre MIDlet3, en el paquete paquete2. Se marcará la opción crear Desplegable, al nombre de la clase se le dará Canvas1 y en el tipo del desplegable se escogerá javax.microedition.lcdui.Canvas.

Resultado(s) esperado(s):

- En el directorio C:/prometeo/pruebas/Proyecto1/src se crea una carpeta de nombre paquete2.
- En el directorio C:/prometeo/pruebas/Proyecto1/src/paquete2 se crean los archivos MIDlet3.java y Canvas1.java
- En el editor aparece el código correspondiente al archivo Canvas1.java
- En el árbol del proyecto aparece el nodo referente al paquete paquete2.

- El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path />
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
  <paquete1.MIDlet2 />
  <paquete2.MIDlet3 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 8: Creación de un MIDlet teniendo asociado un List

Descripción: con el wizard Nuevo MIDlet se creará otro MIDlet de nombre MIDlet4, en el paquete paquete2. Se marcará la opción crear Desplegable, al nombre de la clase se le dará List1 y en el tipo del desplegable se escogerá javax.microedition.lcdui.List

Resultado(s) esperado(s):

- En el directorio C:/prometeo/pruebas/Proyecto1/src/paquete2 se crean los archivos MIDlet4.java y List1.java
- En el editor aparece el código correspondiente al archivo List1.java
- En el árbol, en el nodo correspondiente al paquete2, aparecen los nodos de los archivos MIDlet4.java y List1.java.
- El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path />
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
  <paquete1.MIDlet2 />
  <paquete2.MIDlet3 />
  <paquete2.MIDlet4 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 9: Creación de un MIDlet teniendo asociado un TextBox

Descripción: con el wizard Nuevo MIDlet se creará otro MIDlet de nombre MIDlet5, en el paquete paquete2. Se marcará la opción crear Desplegable, al nombre de la clase se le dará TextBox1 y en el tipo del desplegable se escogerá javax.microedition.lcdui.TextBox

Resultado(s) esperado(s):

- En el directorio C:/prometeo/pruebas/Proyecto1/src/paquete2 se crean los archivos MIDlet5.java y TextBox1.java
- En el editor aparece el código correspondiente al archivo TextBox1.java
- En el árbol, en el nodo correspondiente al paquete2, aparecen los nodos correspondientes a las clases MIDlet5.java y TextBox1.java
- El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path />
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
  <paquete1.MIDlet2 />
  <paquete2.MIDlet3 />
  <paquete2.MIDlet4 />
  <paquete2.MIDlet5 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 10: Creación de una clase para administrar un RecordStore

Descripción: se utilizará el wizard nuevo RecordStore para crear una clase de nombre ControlRecord en un paquete de nombre control, que pueda manejar operaciones sobre un RecordStore de nombre miRecord. miRecord tendrá los siguientes campos:

int cedula (llave primaria)

String Nombre

byte[] foto

boolean estadoCivil

long f_nacimiento

Resultado(s) esperado(s):

- En el directorio C:/Prometeo/Pruebas/Proyecto1/src/ se crea un directorio de nombre control.
- En el directorio C:/Prometeo/Pruebas/Proyecto1/src/control se crea un archivo de nombre ControlRecord.java.
- En el editor aparece el código generado para la clase ControlRecord.java.
- En el árbol aparece el nodo correspondiente al paquete control y dentro de este, el nodo del archivo ControlRecord.java

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados

Prueba No. 11: Creación de una conexión HTTP

Descripción: Se utilizará el wizard Nueva Conexión para crear una clase que se conecte a través de HTTP con un servidor. Al Form que manejará la conexión se le dará el nombre FormConexion y se creará en un paquete de nombre conexión. La dirección de conexión será http://localhost/myserver.

Resultado(s) esperado(s):

- En el directorio C:/Prometeo/Pruebas/Proyecto1/src/ se crea un directorio de nombre conexión.
- En el directorio C:/Prometeo/Pruebas/Proyecto1/src/conexión/ se crean los archivos FormConexion.java y ConexionHTTP.java
- En el editor aparecen las pestañas correspondientes a los archivos FormConexion.java y ConexionHTTP.java.
- En el árbol aparece el nodo correspondiente al paquete conexion y dentro de este los nodos correspondientes a los archivos FormConexion.java y ConexionHTTP.java.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 12: Adición de un Fólder al proyecto

Descripción: Al proyecto Proyecto1 se le adicionará un fólde de nombre miFolder, haciendo uso del menú **Proyecto / Adicionar Fólde**.

Resultado(s) esperado(s):

En el directorio C:/Prometeo/Pruebas/Proyecto1/res debe crearse un directorio de nombre miFolder.

En el árbol del proyecto, en el nodo **res** debe aparecer el nodo correspondiente al directorio miFolder.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 13: Configuración del WTK a un Proyecto

Descripción: Haciendo uso del menú **Herramientas / Configurar WTK** se seleccionará el WTK 2.2 (C:/WTK22) como WTK para el proyecto Proyecto1.

Resultado(s) esperado(s):

El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path>C:\WTK22</path>
  <bibliotecas />
  <emulador>DefaultColorPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
  <paquete1.MIDlet2 />
  <paquete2.MIDlet3 />
  <paquete2.MIDlet4 />
  <paquete2.MIDlet5 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 14: Selección de un Emulador

Descripción: Haciendo uso del menú Herramientas / Configurar Emulador se seleccionará el dispositivo **DefaultGrayPhone** como emulador para el proyecto Proyecto1.

Resultado(s) esperado(s):

El archivo Proyecto1.ppr debe tener el siguiente contenido:

```
Proyecto1.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto1.ppr</nombre>
  <path>C:\WTK22</path>
  <bibliotecas />
  <emulador>DefaultGrayPhone</emulador>
- <midlets>
  <paquete1.MIDlet1 />
  <paquete1.MIDlet2 />
  <paquete2.MIDlet3 />
  <paquete2.MIDlet4 />
  <paquete2.MIDlet5 />
</midlets>
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 15: Adición de un emulador al WTK 2.2

Descripción: Teniendo la carpeta del dispositivo de emulación SonyEricsson_P900, se copiará esta carpeta en el directorio C:\WTK22\wtklib\devices. Después de esto, en prometeo se seleccionará el menú **Herramientas / Configurar Emulador**.

Resultado(s) esperado(s):

En la lista de dispositivos de emulación debe aparecer el dispositivo SonyEricsson_P900

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 16: Actualizar un Proyecto

Descripción: Se copiará el archivo de imagen ok.png en el directorio C:/Prometeo/Pruebas/Proyecto1/res/miFolder. Después de esto, en prometeo se seleccionara el menú **Proyecto / Actualizar**.

Resultado(s) esperado(s):

En el árbol del proyecto, aparece el nodo correspondiente al archivo ok.png en el nodo correspondiente al directorio miFolder.

Resultado(s) obtenidos(s):

Se obtuvo el resultado esperado.

Prueba No. 17: Compilación de un Proyecto

Descripción: Se pulsará el botón compilar teniendo abierto el proyecto Proyecto1

Resultado(s) esperado(s):

- Aparece una barra de progreso indicando el progreso de la compilación.
- Se activa el área de mensajes y en esta la pestaña de compilación.
- En el directorio C:\Prometeo\pruebas\Proyecto1\ se crea un directorio de nombre **classes**.
- Para cada paquete dentro del directorio C:\Prometeo\pruebas\Proyecto1\src se crea el equivalente dentro del directorio C:\Prometeo\pruebas\Proyecto1\classes con los archivos .class producto de la compilación y preverificación.
- Se copia cada uno de los activos y directorios dentro del directorio C:\Prometeo\pruebas\Proyecto1\res en el directorio C:\Prometeo\pruebas\Proyecto1\classes
- Aparece el mensaje “→ Compilación Completa” en la pestaña de compilación del área de mensajes.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados, sin embargo, en otras pruebas realizadas en un proyecto cuya dirección de ubicación en el sistema de archivos contenía espacios, sobre la plataforma Linux no se compilaron los archivos java.

Observaciones:

Los nombres con espacios en Windows no generaron mayor problema, basta con adicionar comillas ("") a las distintas rutas, sin embargo en Linux (Mandrake 9.1 y Slackware 10) las comillas eran reconocidas como parte del nombre del archivo, por lo cual no era encontrado. Dado lo anterior, una de las restricciones de Prometeo, al menos por ahora, es que sobre la plataforma Linux las rutas de los proyectos creados no deben contener espacios.

Prueba No. 18: Ejecución de un Proyecto

Descripción: Se pulsará el botón Ejecutar, estando abierto el proyecto Proyecto1.

Resultado(s) esperado(s):

- Se activa el área de mensajes y en esta la pestaña de ejecución y en esta aparece el comando:

```
C:\WTK22\bin\emulator -classpath "C:\Prometeo\pruebas\Proyecto1\classes;" -Xdevice:DefaultGrayPhone -Xdescriptor:"C:\Prometeo\pruebas\Proyecto1\bin\Proyecto1.jad"
```
- Se lanza el emulador con el dispositivo DefaultGrayPhone (escogido anteriormente) y en él se muestra una lista con los nombres de los MIDlets uno hasta cinco.
- Al terminar la ejecución se mostrará en la pestaña de ejecución las características de ejecución de Proyecto1.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Observaciones:

Sobre la plataforma Linux, debido a que no se pudieron compilar proyectos con espacios en sus paths, tampoco se puede hacer la ejecución.

Prueba No. 19: Compilación de un archivo con errores (falta de un ;).

Descripción: En Proyecto1 se creará una clase de nombre ErrorClass en el paquete paquete1. Dado que el código generado se hace sin errores, se borrará el “;” de la primera línea de la clase. Se procederá a compilar el Proyecto.

Resultado(s) esperado(s):

En la pestaña de compilación aparece el error correspondiente indicando el archivo donde ocurrió, la línea y el tipo de error.

Resultado(s) obtenidos(s):

- Utilizando el J2SDK 1.4.1_01 se obtuvieron exactamente los resultados esperados.
- Utilizando el J2SDK 1.4.2_04 se muestra el mensaje en la pestaña de compilación pero como si el error estuviera en la línea 2 del archivo ErrorClass.java

Observaciones:

La ambigüedad presentada es propia de la utilización de una u otra máquina virtual de java y está por fuera del alcance de este trabajo.

Prueba No. 20: Compilación de un proyecto con errores (referencia desconocida)

Descripción: En la clase ErrorClass se adicionará el “;” en la primera línea, a continuación, en la sección de atributos se adicionará la línea “Algo a;”. Se procederá a compilar el proyecto.

Resultado(s) esperado(s):

En la pestaña de compilación aparece el error correspondiente indicando el archivo donde ocurrió, la línea y el tipo de error.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Observaciones:

Los resultados se dieron de igual forma indistintamente de la JVM utilizada.

Prueba No. 21: Eliminación de un archivo

Descripción: En el árbol del proyecto se dará clic derecho sobre el nodo correspondiente al archivo ok.png, se escogerá la opción Eliminar “ok.png”.

Resultado(s) esperado(s):

- Aparece un cuadro de dialogo para confirmar que el archivo ok.png será eliminado.

- El archivo ok.png es eliminado del directorio C:\Prometeo\pruebas\Proyecto1\res\miFolder
- El directorio miFolder es eliminado dado que queda vacío.
- El nodo correspondiente al archivo ok.png es eliminado del árbol del proyecto.

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Observaciones:

Los archivos no se eliminan del directorio classes. Para actualizar este directorio es necesario compilar el proyecto.

Prueba No. 22: Adicionar una biblioteca de clases a PROMETEO

Descripción: se escogerá el menú Herramientas / Configurar Bibliotecas para adicionar una biblioteca de nombre Biblioteca1. El path de la biblioteca será C:\Prometeo\pruebas\Proyecto1\classes\conexion.

Resultado(s) esperado(s):

El archivo config.XML debe tener el siguiente contenido:

Config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
- <prometeo>
- <librerias>
  - <Biblioteca1>
    <path0>C:\Prometeo\pruebas\Proyecto1\classes\conexion</path0>
  </Biblioteca1>
</librerias>
<ultimoproyecto>C:\Prometeo\pruebas\Proyecto1\Proyecto1.ppr</ultimoproyecto>
</prometeo>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 23: Adicionar una biblioteca a un Proyecto

Descripción: se cierra Proyecto1 y en el directorio C:\Prometeo\Pruebas se crea el proyecto Proyecto2. Haciendo uso del menú Proyecto / Propiedades del Proyecto se adiciona la biblioteca Biblioteca1 a Proyecto2

Resultado(s) esperado(s):

- Se crean todos los directorios y archivos propios de la creación de un Proyecto.
- El archivo Proyecto2.ppr debe tener el siguiente contenido:

```
Proyecto2.ppr
<?xml version="1.0" encoding="UTF-8" ?>
- <proyecto>
  <nombre>Proyecto2.ppr</nombre>
  <path />
  <bibliotecas>Biblioteca1-</bibliotecas>
  <emulador>DefaultColorPhone</emulador>
  <midlets />
  <paquetes />
</proyecto>
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 24: Compilación de un proyecto con una biblioteca asociada

Descripción: para comprobar que el path de la biblioteca es tomado al momento de la compilación del Proyecto2, se creará un MIDlet de nombre MIDletP2 sin desplegable asociado en un paquete de nombre paquete1. En la sección de atributos de MIDletP2.java se adicionará la línea “`conexion.ConexionHttp c;`”. Se seleccionará el WTK22 para Proyecto2 y se procederá a compilar el Proyecto.

Resultado(s) esperado(s):

- En la pestaña de compilación en el área de mensajes aparece el mensaje “→Compilación Completa” y ningún error.
- En el directorio C:\Prometeo\pruebas\Proyecto2\classes aparece el paquete paquete1 y en su interior el archivo MIDletP2.class

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 25: Crear un archivo JAR

Descripción: haciendo uso del menú Herramientas / Crear se crea un JAR para Proyecto2. Se dejarán las opciones por defecto que ofrece PROMETEO.

Resultado(s) esperado(s):

- En el directorio C:\Prometeo\pruebas\Proyecto2\bin se crea un archivo de nombre MANIFIESTO.MF con el siguiente contenido:

MANIFIESTO.MF

```
Manifest-Version: 1.0
MIDlet-Version: 1.0
MIDlet-1: MIDletP2, ,paquete1.MIDletP2
MIDlet-Name: MySuite
MIDlet-Vendor: Prometeo
MIDlet-Icon: /MyIcon
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-2.0
```

- En el directorio C:\Prometeo\pruebas\Proyecto2\bin se crea un archivo de nombre Proyecto2.jar con el siguiente contenido:

Name	Path
Manifest.mf	meta-inf\
MIDletP2.class	paquete1\
FormConexion.class	conexion\
ConexionHttp.class	conexion\

- En el directorio C:\Prometeo\pruebas\Proyecto2\bin se crea un archivo de nombre Proyecto2.jad con el siguiente contenido:

MANIFIESTO.MF

```
MIDlet-1: MIDletP2, ,paquete1.MIDletP2
MIDlet-Name: MySuite
MIDlet-Vendor: Prometeo
MIDlet-Version: 1.0
MIDlet-Jar-Size: 3919
MIDlet-Jar-URL: Proyecto2.jar
```

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.

Prueba No. 26: Desvincular una Biblioteca de un Proyecto

Descripción: haciendo uso del menú Proyecto / Propiedades del Proyecto se removerá la biblioteca Biblioteca1 de Proyecto2. Para comprobar que Biblioteca1 ya no hace parte del path de Proyecto2, se procederá a compilar el proyecto.

Resultado(s) esperado(s):

- En la pestaña de compilación aparece un mensaje relacionado con el paquete conexión (no existe el paquete).
- No se copia ningún archivo en el directorio C:\Prometeo\Pruebas\Proyecto2\Clases

Resultado(s) obtenidos(s):

Se obtuvieron los resultados esperados.