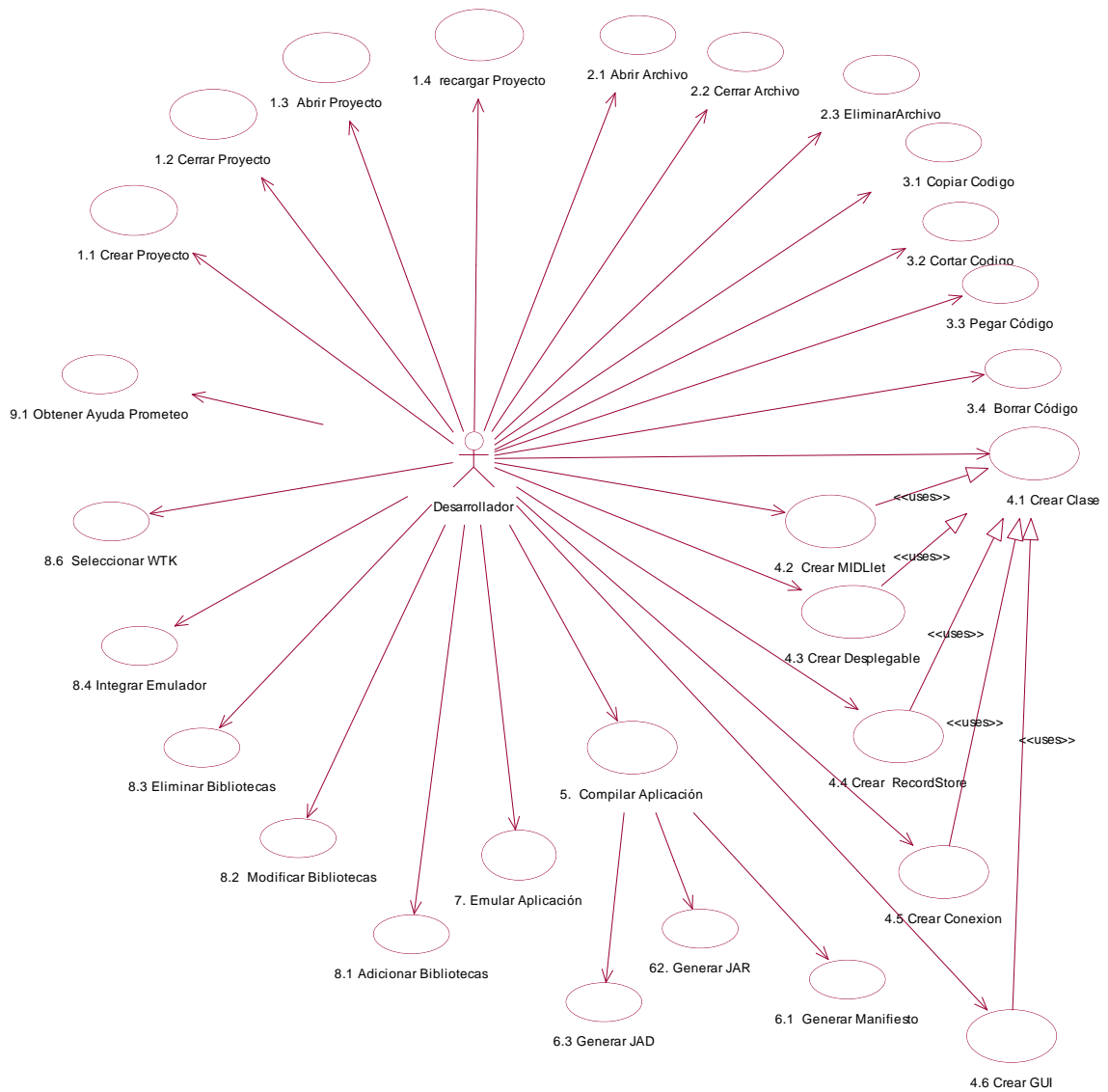


## CONTENIDO

DIAGRAMA DE CASOS DE USO .....	2
CASO DE USO No. 2: CERRAR PROYECTO .....	5
CASO DE USO No. 3: ABRIR PROYECTO .....	7
CASO DE USO No. 4: ACTUALIZAR PROYECTO .....	9
CASO DE USO No. 5: ABRIR ARCHIVO .....	11
CASO DE USO No. 6: CERRAR ARCHIVO .....	13
CASO DE USO No. 7: ELIMINAR ARCHIVO .....	14
CASO DE USO No. 8: COPIAR CÓDIGO .....	16
CASO DE USO No. 9: CORTAR CÓDIGO .....	17
CASO DE USO No. 10: PEGAR CÓDIGO .....	18
CASO DE USO No. 11: CREAR CLASE .....	19
CASO DE USO No. 12: CREAR MIDLET .....	21
CASO DE USO No. 13: CREAR DESPLEGABLE .....	23
CASO DE USO No. 14: CREAR CONEXIÓN .....	25
CASO DE USO No. 15: CREAR RECORD STORE .....	27
CASO DE USO No. 16: CREAR GUI .....	29
CASO DE USO No. 17: COMPILAR APLICACIÓN .....	31
CASO DE USO No. 18: CREAR JAR .....	33
CASO DE USO No. 19: GENERAR MANIFIESTO .....	35
CASO DE USO No. 20: GENERAR JAD .....	37
CASO DE USO No. 21: EMULAR APLICACIÓN .....	39
CASO DE USO No. 22: ADICIONAR BIBLIOTECAS .....	41
CASO DE USO No. 23: MODIFICAR BIBLIOTECAS .....	43
CASO DE USO No. 24: ELIMINAR BIBLIOTECAS .....	44
CASO DE USO No. 25: ADICIONAR EMULADOR .....	45
CASO DE USO No. 26: SELECCIONAR EMULADOR .....	46
CASO DE USO No. 27: CONFIGURAR WTK .....	47
CASO DE USO No. 28: OBTENER AYUDA DEL ENTORNO .....	49
CASO DE USO No. 29: OBTENER AYUDA DE J2ME .....	49
DIAGRAMA DE CLASES .....	49

# DISEÑO DETALLADO

## DIAGRAMA DE CASOS DE USO



## CASO DE USO No. 1: CREAR PROYECTO

**Actores:** Desarrollador (iniciador)

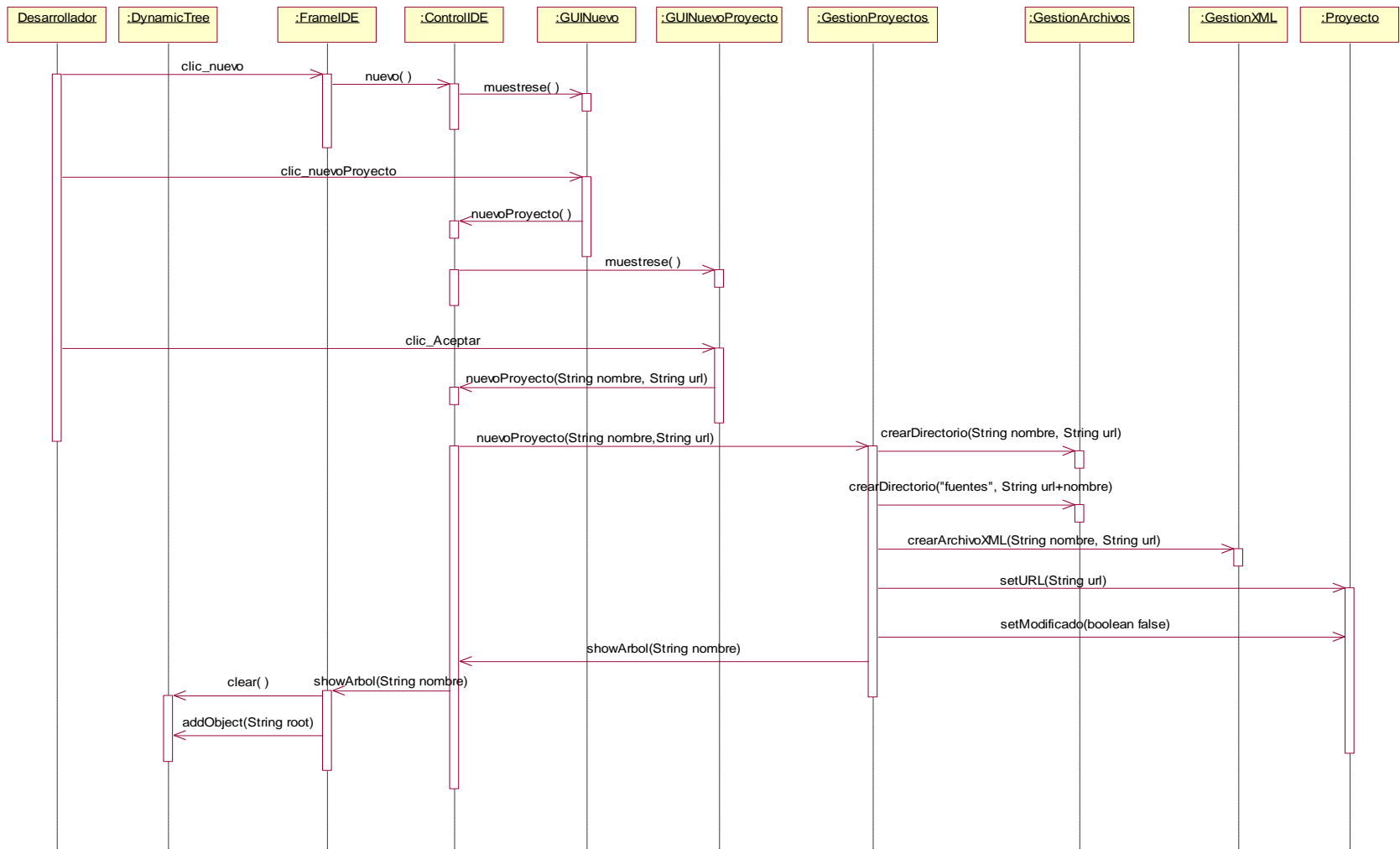
**Propósito:** Permitir al desarrollador la creación de un directorio en donde se almacenen todos los recursos necesarios para su aplicación (código fuente, código objeto, archivos XML, JAR, JAD, imágenes, etc).

**Resumen:** El desarrollador proporciona el nombre y la ubicación del proyecto en el sistema de archivos. El sistema crea una carpeta con el nombre del proyecto y le muestra al desarrollador el nombre del nuevo proyecto en la GUI principal.

**Flujo principal:**

- Este caso de uso empieza cuando el desarrollador hace clic sobre el icono *Nuevo* o cuando selecciona en el menú la opción *Archivo / Nuevo*.
- El sistema responde presentando la GUI *Nuevo*
- El desarrollador hace clic sobre la opción *Proyecto*.
- El sistema responde mostrando la GUI *Nuevo Proyecto*
- El desarrollador proporciona el nombre y la ubicación del proyecto.
- El desarrollador selecciona *Aceptar*.
- El sistema crea una carpeta con el nombre del proyecto, dentro de esta carpeta se crea un archivo con extensión ".ppr" (proyecto de prometeo) y el nombre dado por el desarrollador para su proyecto (este archivo es un documento XML que sirve para almacenar de manera persistente características de configuración de cada proyecto creado), finalmente el sistema muestra en la GUI principal el nombre del proyecto seguido de la extensión ".ppr", concluyendo de esta forma el caso de uso.

### DIAGRAMA DE SECUENCIA



## CASO DE USO No. 2: CERRAR PROYECTO

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador cerrar el proyecto abierto en el entorno.

**Resumen:** El desarrollador selecciona la opción “cerrar nombre\_proyecto.ppr” y confirma la acción.

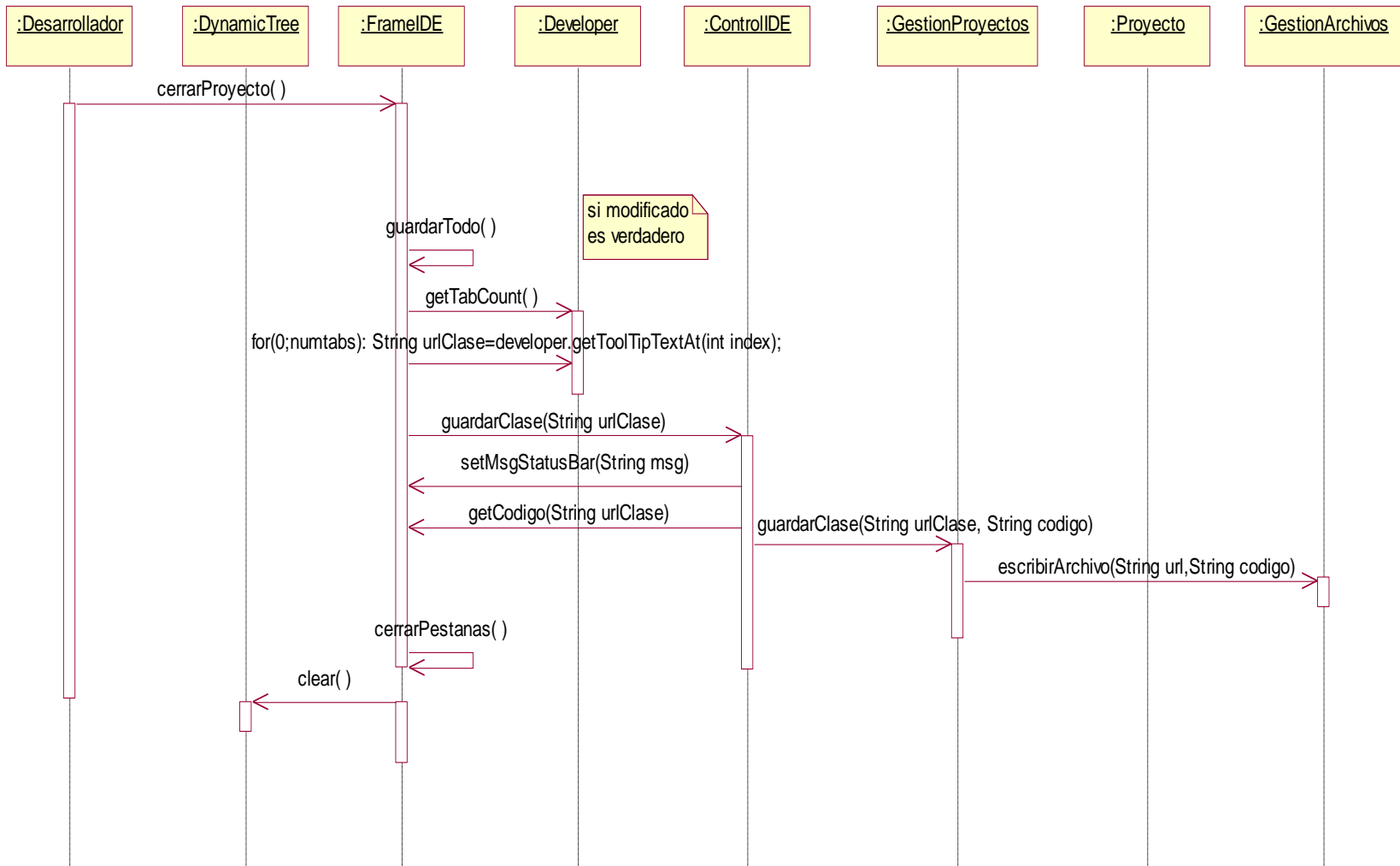
**Precondiciones:**

Debe haber un proyecto abierto en el entorno.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador hace clic derecho sobre el nombre de su proyecto en el árbol que muestra la estructura de archivos de su aplicación y selecciona en el menú emergente la opción “cerrar nombre\_proyecto.ppr”.
- El sistema responde presentando una ventana de confirmación en la que se le pregunta al desarrollador si desea cerrar el proyecto guardando o no los cambios.
- Si el desarrollador selecciona si, el sistema responde guardando todos los archivos de la aplicación y reiniciando todo el entorno, terminando así el caso de uso.
- Si el desarrollador selecciona no, el sistema responde reiniciando todo el entorno, terminando así el caso de uso.

**DIAGRAMA DE SECUENCIA**



### **CASO DE USO No. 3: ABRIR PROYECTO**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador abrir un proyecto creado para continuar trabajando en el desarrollo de su aplicación.

**Resumen:** El desarrollador selecciona abrir en la GUI principal y proporciona la dirección del archivo .ppr del proyecto que desea abrir.

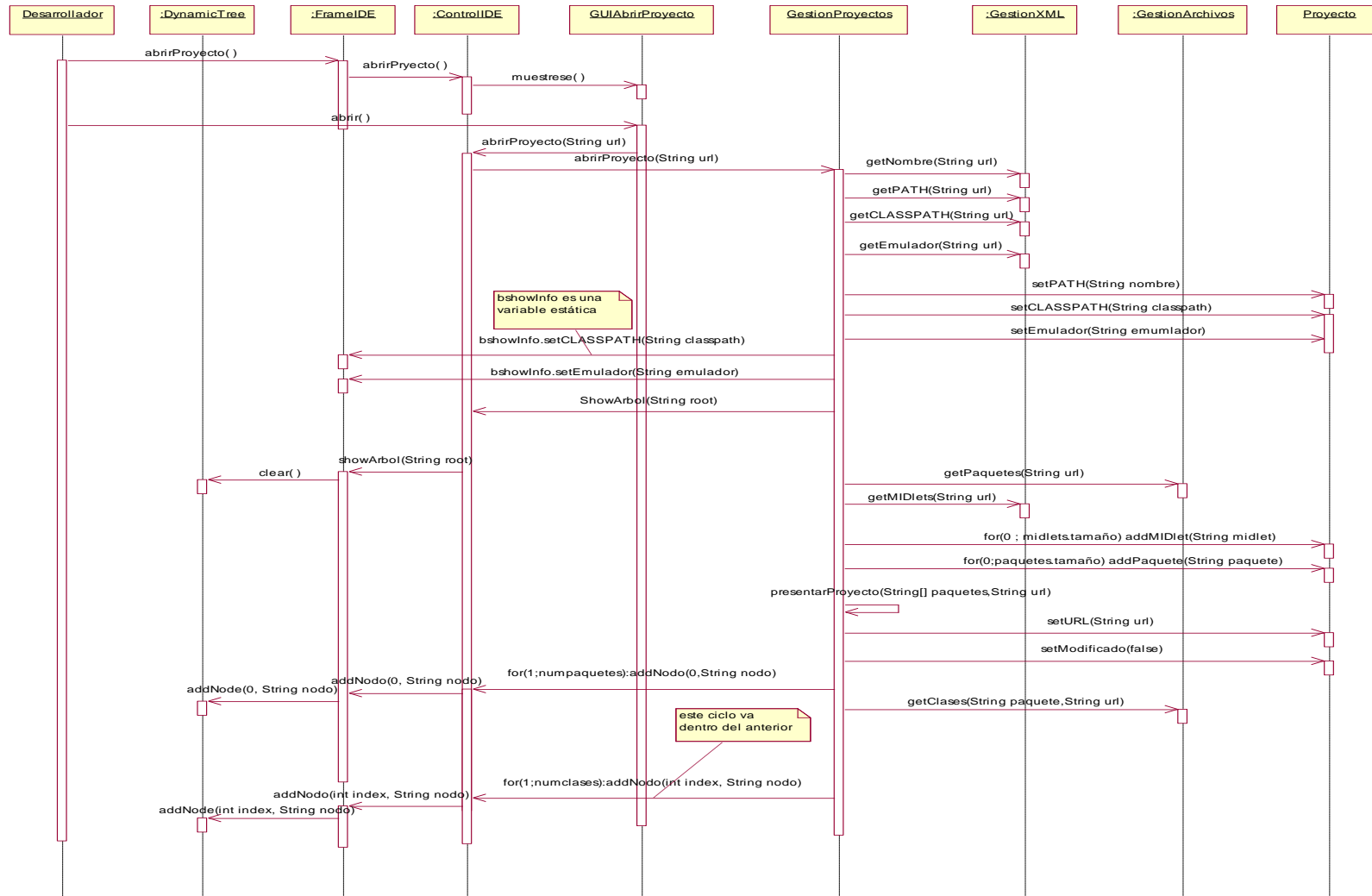
**Precondiciones:**

Debe existir el archivo .ppr del proyecto que el desarrollador desee abrir en el entorno.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el icono “abrir” o selecciona en el menu la opción Archivo / Abrir.
- El sistema responde presentando una GUI en la que el desarrollador puede buscar y seleccionar el archivo ppr del proyecto que desee abrir
- El usuario pulsa el botón “Abrir”.
- El sistema responde mostrando en la GUI principal la estructura de archivos del proyecto abierto, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA





**CASO DE USO No. 4: ACTUALIZAR PROYECTO**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador visualizar la estructura de archivos actual de su aplicación cuando haga algún cambio en alguno de los directorios.

**Resumen:** El desarrollador selecciona “actualizar” en la GUI.

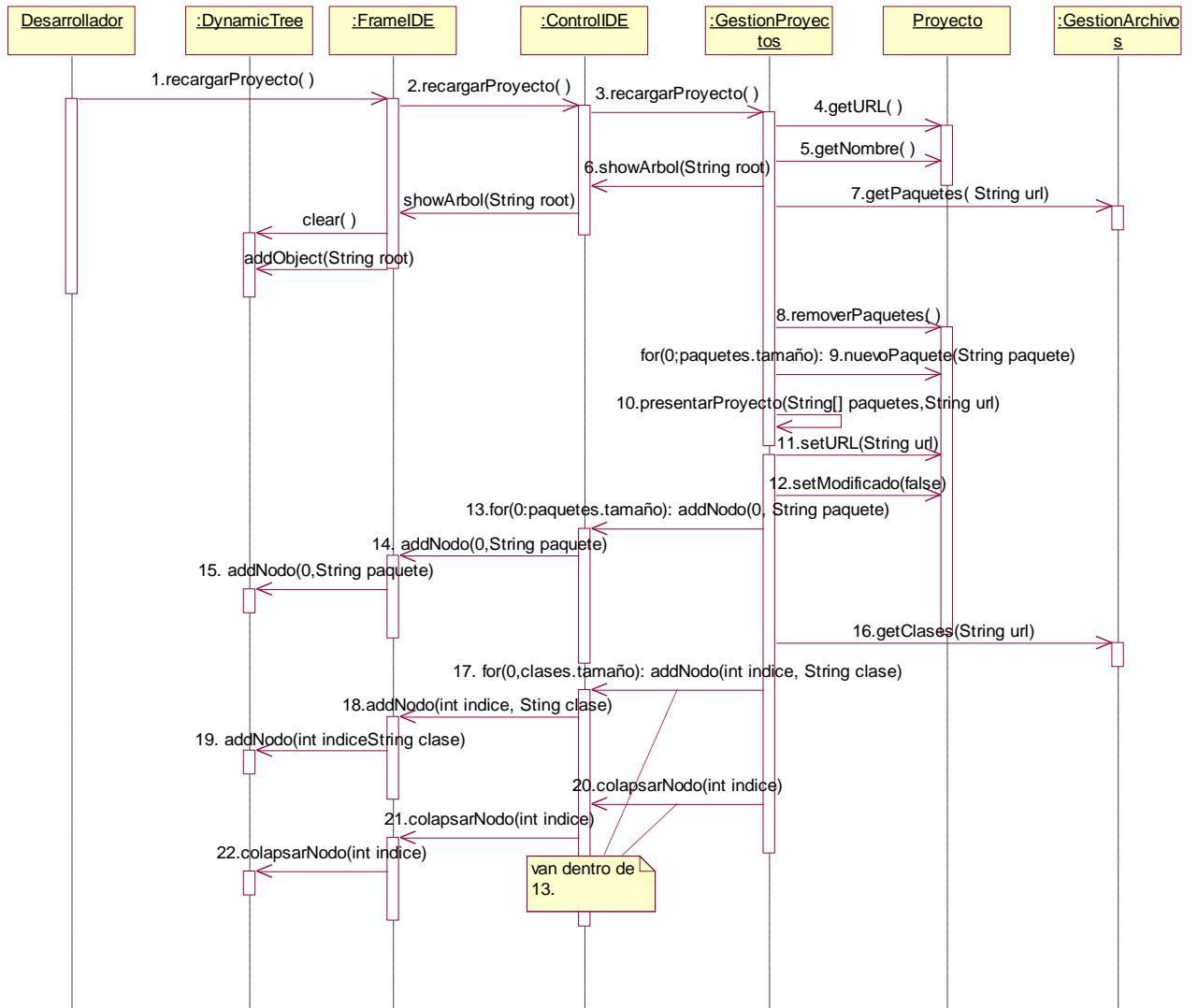
**Precondiciones:**

- Debe haber un proyecto abierto en el entorno.
- Puede iniciarse desde el caso de uso Eliminar Archivo.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador hace clic sobre el icono “actualizar” o selecciona en el menú la opción Proyecto / Actualizar.
- El sistema responde mostrando la estructura de archivos actualizada del proyecto sobre el cual el desarrollador está trabajando, finalizado así el caso de uso.

DIAGRAMA DE SECUENCIA



**CASO DE USO No. 5: ABRIR ARCHIVO**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador abrir diferentes tipos de archivos en el entorno de desarrollo, particularmente archivos java, xml, txt y wsdl.

**Resumen:** El desarrollador selecciona el archivo que desea abrir y elige la opción “abrir” o da doble clic sobre el archivo en el árbol que muestra la estructura de archivos de su proyecto.

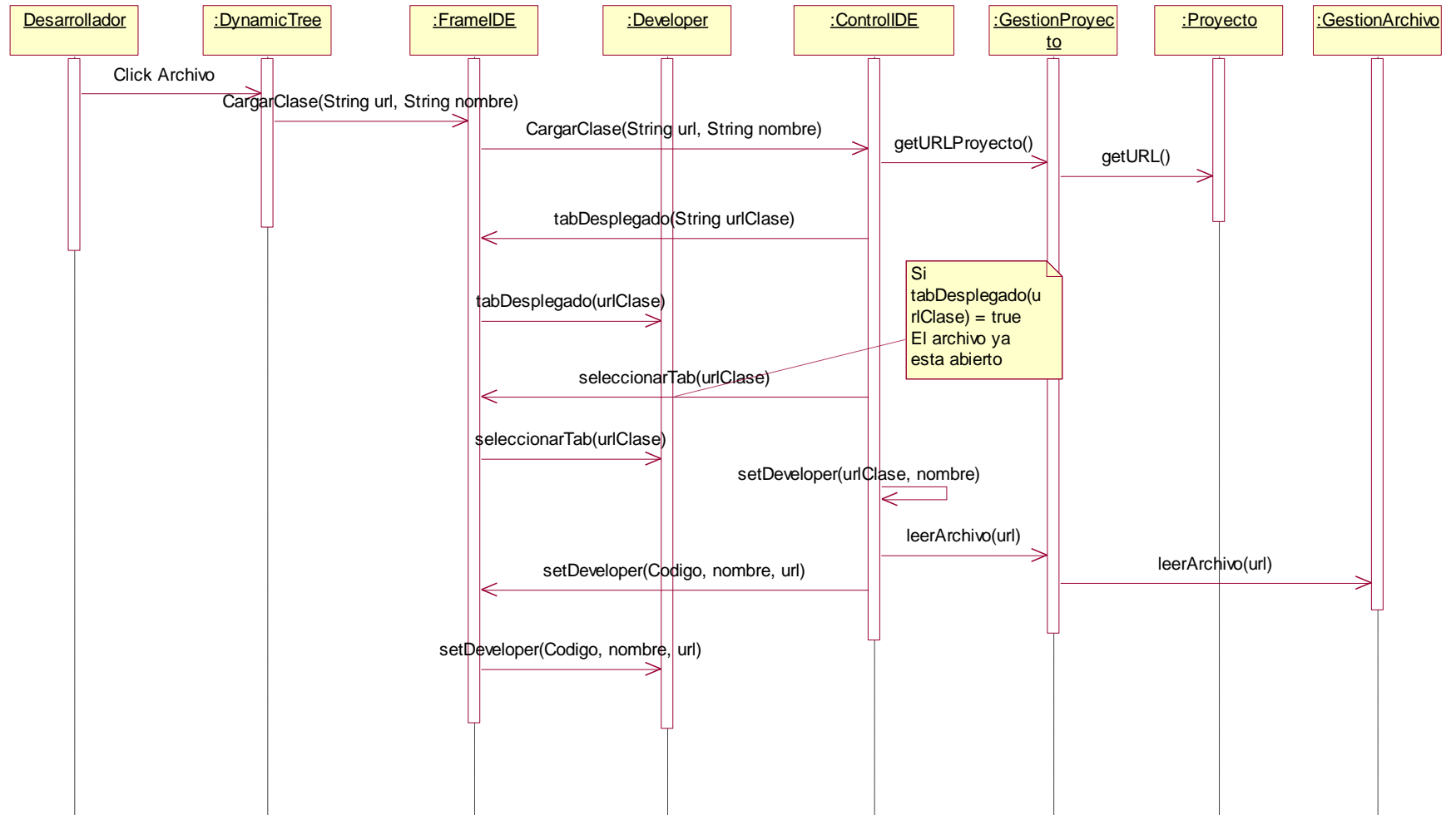
**Precondiciones:**

- El archivo que el desarrollador desee abrir en el entorno debe existir y debe tener una de las siguientes extensiones: java, xml, txt o wsdl.
- El archivo debe verse en el árbol que muestra la estructura de archivos.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador hace doble clic sobre uno de los archivos en el árbol que muestra la estructura de archivos de su proyecto o cuando hace clic derecho sobre uno de estos archivos y en el menú emergente selecciona *Abrir nombre\_archivo*.
- El sistema responde mostrando en el editor de código una pestaña con el nombre del archivo abierto y el contenido del mismo, finalizando así el caso de uso.

### DIAGRAMA DE SECUENCIA



## CASO DE USO No. 6: CERRAR ARCHIVO

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador finalizar la visualización de un archivo que este abierto en el entorno.

**Resumen:** el desarrollador selecciona el archivo que desea cerrar y elige la opción “cerrar nombre\_archivo”.

**Precondiciones:**

El archivo a cerrar debe estar abierto en el editor de código.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador hace clic derecho sobre la pestaña que representa el archivo que desea cerrar.
- El sistema responde mostrando un menú emergente con las opciones de cerrar el archivo seleccionado, cerrar todos los demás archivos abiertos o cerrar todos los archivos.
- Si el desarrollador selecciona “cerrar nombre\_archivo” el sistema responde cerrando esa pestaña en el editor, finalizando así el caso de uso.
- Si el desarrollador selecciona “cerrar otros archivos” el sistema responde cerrando todas las pestañas que se encuentren abiertas en el editor, con excepción de la que seleccionó el desarrollador, finalizando de esta forma este caso de uso.
- Si el desarrollador selecciona “cerrar todos los archivos” el sistema responde cerrando todas las pestañas que se encuentren abiertas en el editor, finalizando así el caso de uso.

**CASO DE USO No. 7: ELIMINAR ARCHIVO**

**Actores:** Desarrollador (iniciador)

**Propósito:** brindar al desarrollador la capacidad para eliminar desde el entorno los archivos que no sean requeridos para el desarrollo de su aplicación.

**Resumen:** El desarrollador selecciona el archivo que desea eliminar, elige la opción *Eliminar* y confirma al sistema que desea eliminar el archivo seleccionado.

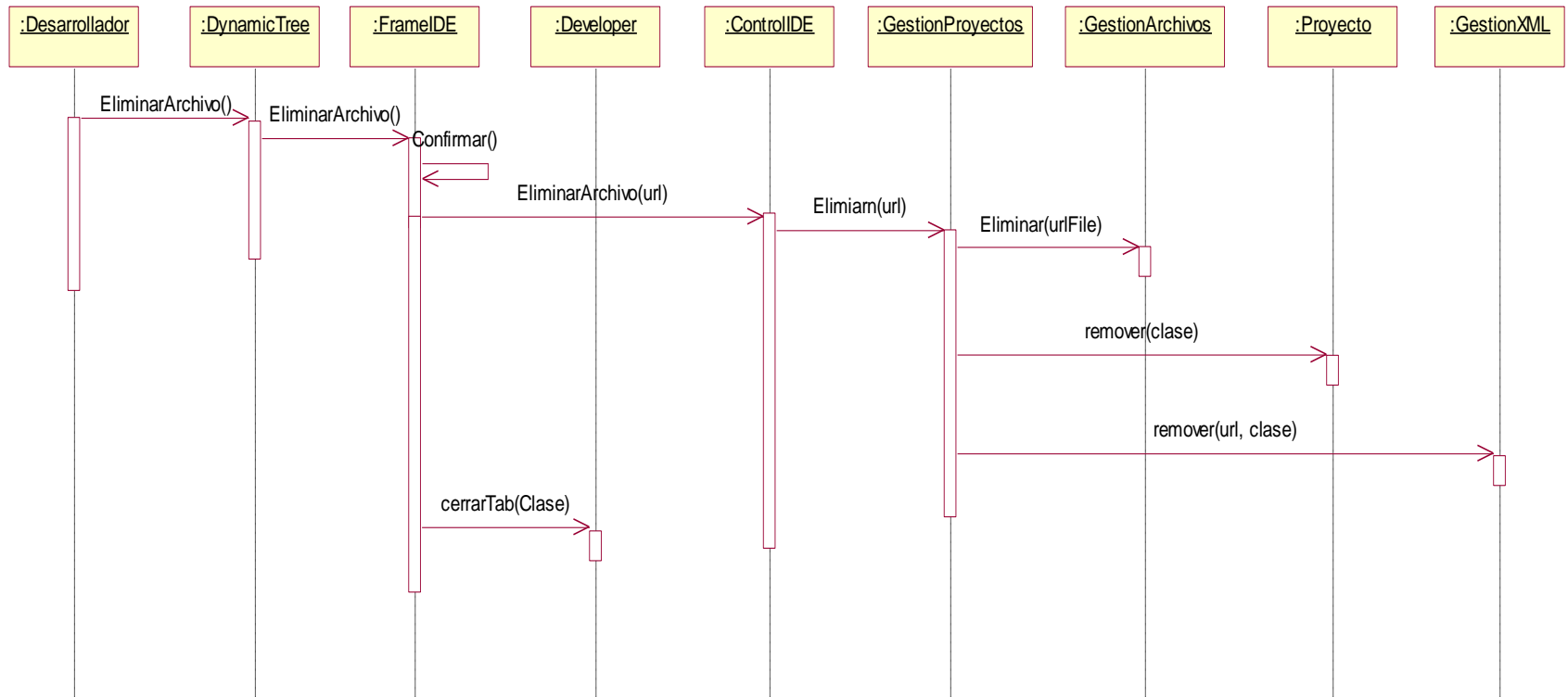
**Precondiciones:**

- Debe estar abierto un proyecto en el entorno.
- El archivo que se desea eliminar debe ser visible en el árbol de la estructura de archivos de la aplicación.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic derecho sobre el nodo en el árbol que representa al archivo que desea eliminar y en el menú emergente hace clic en la opción "Eliminar nombre\_archivo".
- El sistema responde mostrando una ventana de confirmación en la que se pregunta al desarrollador si en realidad desea eliminar el archivo seleccionado
- Si el desarrollador selecciona si el sistema responde eliminando el archivo seleccionado y actualizando el árbol de la estructura de archivos del proyecto, finalizando así el caso de uso.
- Si el desarrollador selecciona no, el sistema responde ocultando la ventana de confirmación, finalizando así el caso de uso.

### DIAGRAMA DE SECUENCIA



## CASO DE USO No. 8: COPIAR CÓDIGO

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador copiar código y en general texto entre archivos abiertos en el entorno y desde archivos abiertos en el entorno hacia otros archivos ajenos al entorno y viceversa.

**Resumen:** el desarrollador selecciona el código o texto que desee copiar, elige la opción copiar.

**Precondiciones:**

Debe haber un archivo abierto en el editor de código del entorno.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador selecciona el código o texto que desea copiar, pulsa clic derecho y en el menú emergente selecciona la opción *copiar* o cuando el desarrollador selecciona el código o texto que desea copiar y pulsa las teclas CTRL + C. O cuando el desarrollador hace clic en el icono *copiar*.
- El sistema responde almacenando en el clipboard sistema operativo la selección hecha por el usuario finalizando así el caso de uso.

Nota: Como el contenido seleccionado está en el portapapeles del sistema operativo, este contenido puede copiarse a cualquier parte permitida por el sistema operativo.



## CASO DE USO No. 9: CORTAR CÓDIGO

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador cortar código o en general texto desde un archivo abierto en el entorno.

**Resumen:** el desarrollador selecciona el código o texto que desea cortar y hace clic en la opción cortar.

**Precondiciones:**

El archivo del que se desee cortar código debe estar abierto en el entorno.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador selecciona el código o texto que desea cortar, hace clic derecho sobre el editor y en el menú emergente selecciona la opción *cortar*. O cuando habiendo seleccionado el código o texto pulsa las teclas CTRL + X o hace clic en el icono *cortar*.
- El sistema responde eliminando el código o texto seleccionado del editor y almacenándolo en el portapapeles del sistema operativo, finalizando así el caso de uso.

**CASO DE USO No. 10: PEGAR CÓDIGO**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador pegar el código o texto que esté almacenado en la memoria del sistema operativo.

**Resumen:** El desarrollador se ubica en el archivo en el que desee pegar el código o texto y selecciona la opción *pegar*.

**Precondiciones:**

- Debe haber un archivo abierto en el entorno.
- Debe haber código o texto en el clipboard del sistema operativo.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic derecho en el editor y en el menú emergente elige la opción *pegar*. O cuando pulsa el icono *cortar* o pulsa las teclas CTRL. + V.
- El sistema responde pegando el código o texto que se encuentra en el clipboard del sistema operativo en la ubicación actual del cursor en el editor, finalizando así el caso de uso.

**CASO DE USO No. 11: CREAR CLASE**

**Actores:** Desarrollador (iniciador)

**Propósito:** brindar al desarrollador un generador de código que le permita crear la plantilla de una nueva clase.

**Resumen:** El desarrollador selecciona la opción crear clase, proporciona la información necesaria (nombre de la clase, paquete) y si desea la información opcional (autor, extiende de, implementa a) y pulsa aceptar.

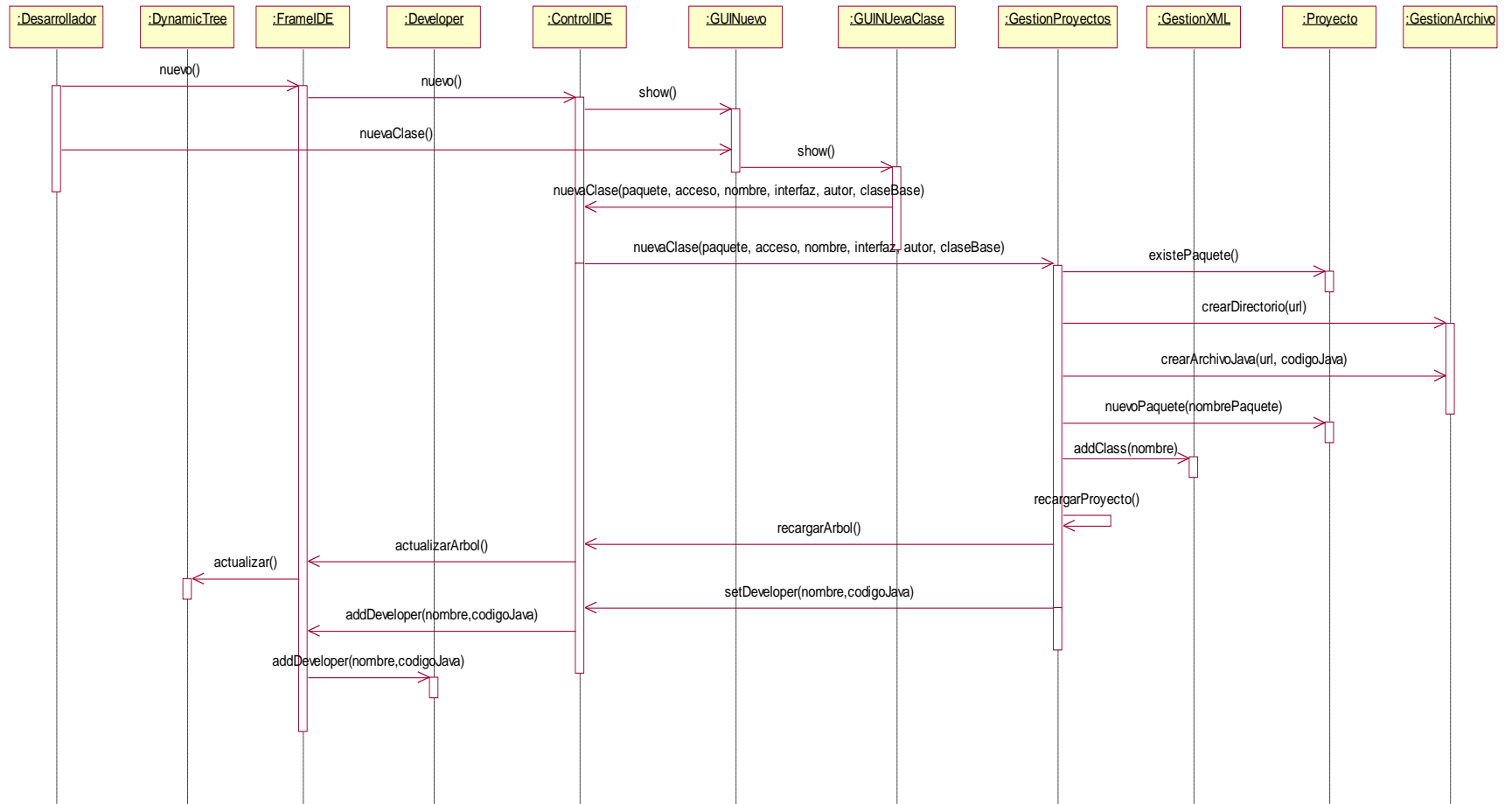
**Precondiciones:**

- Debe haber un proyecto abierto.
- El sistema debe contar con información obligatoria provista por el desarrollador (Nombre de la clase y paquete).

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el icono Nuevo o escoge la opción Archivo / Nuevo en el menú principal y hace clic sobre el botón Clase
- El sistema responde mostrando una GUI dE Nueva clase
- El desarrollador suministra la información necesaria y pulsa el botón Aceptar.
- El sistema responde creando un archivo .java con el nombre y en el paquete dados por el usuario, el cual es mostrado en el editor de código, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA



**CASO DE USO No. 12: CREAR MIDLET**

**Actores:** Desarrollador (iniciador)

**Propósito:** brindar al desarrollador un generador de código que le permita crear la plantilla de un nuevo MIDlet con los métodos y atributos requeridos para que se pueda ejecutar en un emulador.

**Resumen:** El desarrollador selecciona la opción crear MIDlet, proporciona la información necesaria y pulsa aceptar.

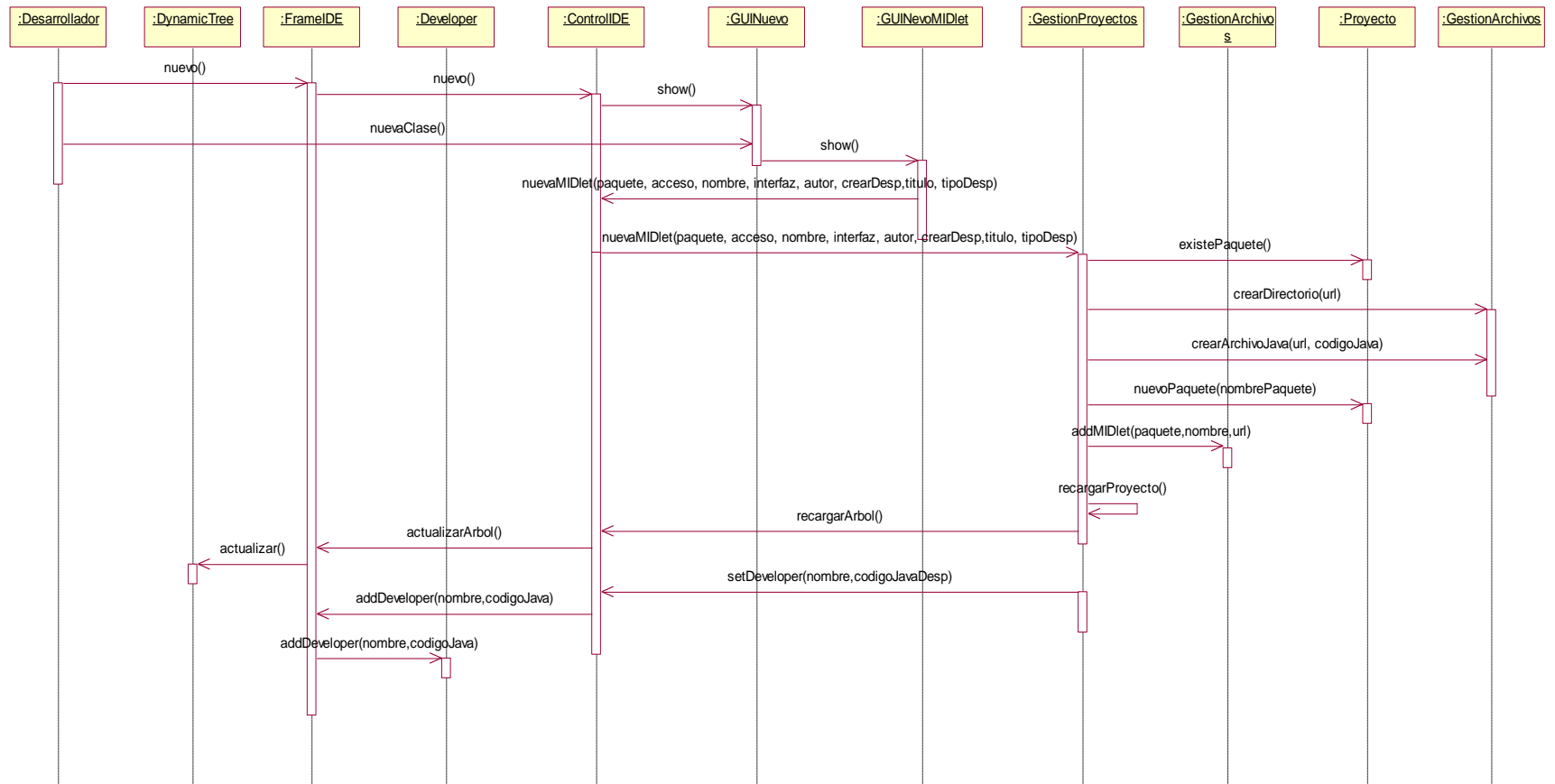
**Precondiciones:**

- Debe haber un proyecto abierto.
- El sistema debe contar con información obligatoria provista por el desarrollador (Nombre de la clase y paquete).

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el icono Nuevo o escoge la opción Archivo / Nuevo en el menú principal y hace clic sobre el botón MIDlet .
- El sistema responde mostrando una GUI como la de la figura 3.10.
- El desarrollador suministra la información necesaria y pulsa el botón Aceptar.
- Si el desarrollador marcó la opción crear Desplegable, el sistema responde creando dos archivos .java, uno para el MIDlet y otro para el Desplegable, con los nombres y en el paquete especificado por el desarrollador y mostrando en el editor el archivo correspondiente al Desplegable, finalizando así el caso de uso.
- Si el desarrollador no marcó la opción crear Desplegable, el sistema responde creando un archivo .java para el nuevo MIDlet con el nombre de clase y en el paquete dados por el usuario, dicho archivo es mostrado en el editor de código, finalizando de esta forma este caso de uso.

DIAGRAMA DE SECUENCIA



**CASO DE USO No. 13: CREAR DESPLEGABLE**

**Actores:** Desarrollador (iniciador)

**Propósito:** brindar la posibilidad de utilizar generadores de código que le permitan crear diferentes tipos de clases para presentar las GUI en el dispositivo móvil (Form, Canvas, List, TextBox).

**Resumen:** El desarrollador selecciona la opción nuevo desplegable, ingresa la información requerida, escoge el tipo de desplegable que desea crear y pulsa el botón aceptar.

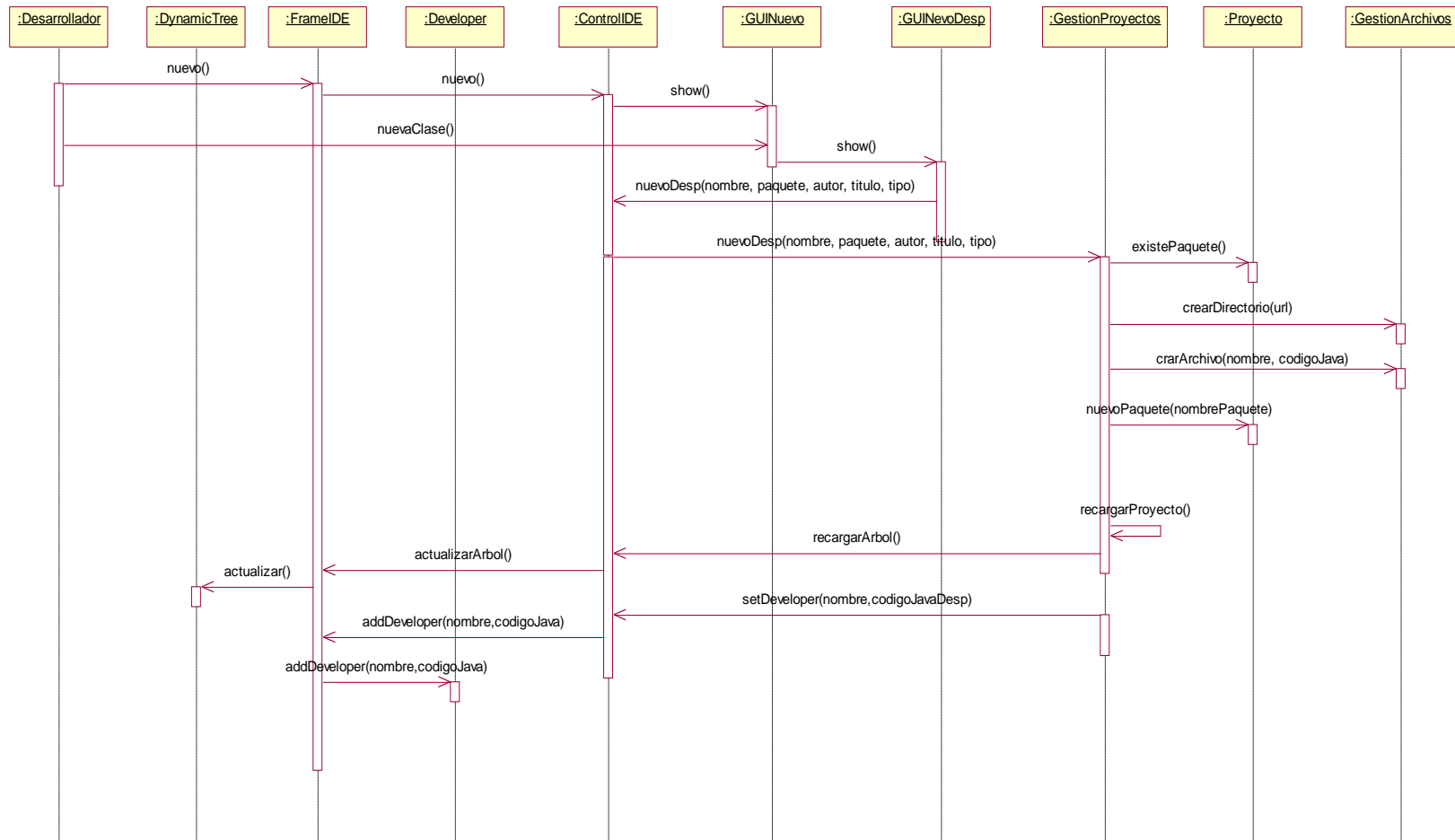
**Precondiciones:**

- Debe haber un proyecto abierto.
- El sistema debe contar con información obligatoria provista por el desarrollador (Nombre de la clase y paquete).

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el icono Nuevo o escoge la opción Archivo / Nuevo en el menú principal y hace clic sobre el botón Desplegable.
- El sistema responde mostrando una GUI NuevoDesplegable
- El desarrollador suministra la información necesaria y pulsa el botón Aceptar.
- El sistema responde creando una clase con el nombre, paquete y tipo de desplegable dados por el desarrollador y mostrando en el editor el archivo correspondiente al desplegable creado, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA





## CASO DE USO No. 14: CREAR CONEXIÓN

**Actores:** Desarrollador (iniciador)

**Propósito:** Permitir al desarrollador la rápida creación de un par de clases que le permitan comunicarse con una red de datos a través de una conexión HTTP.

**Resumen:** El desarrollador selecciona la opción nueva conexión, suministra la información requerida y pulsa el botón aceptar.

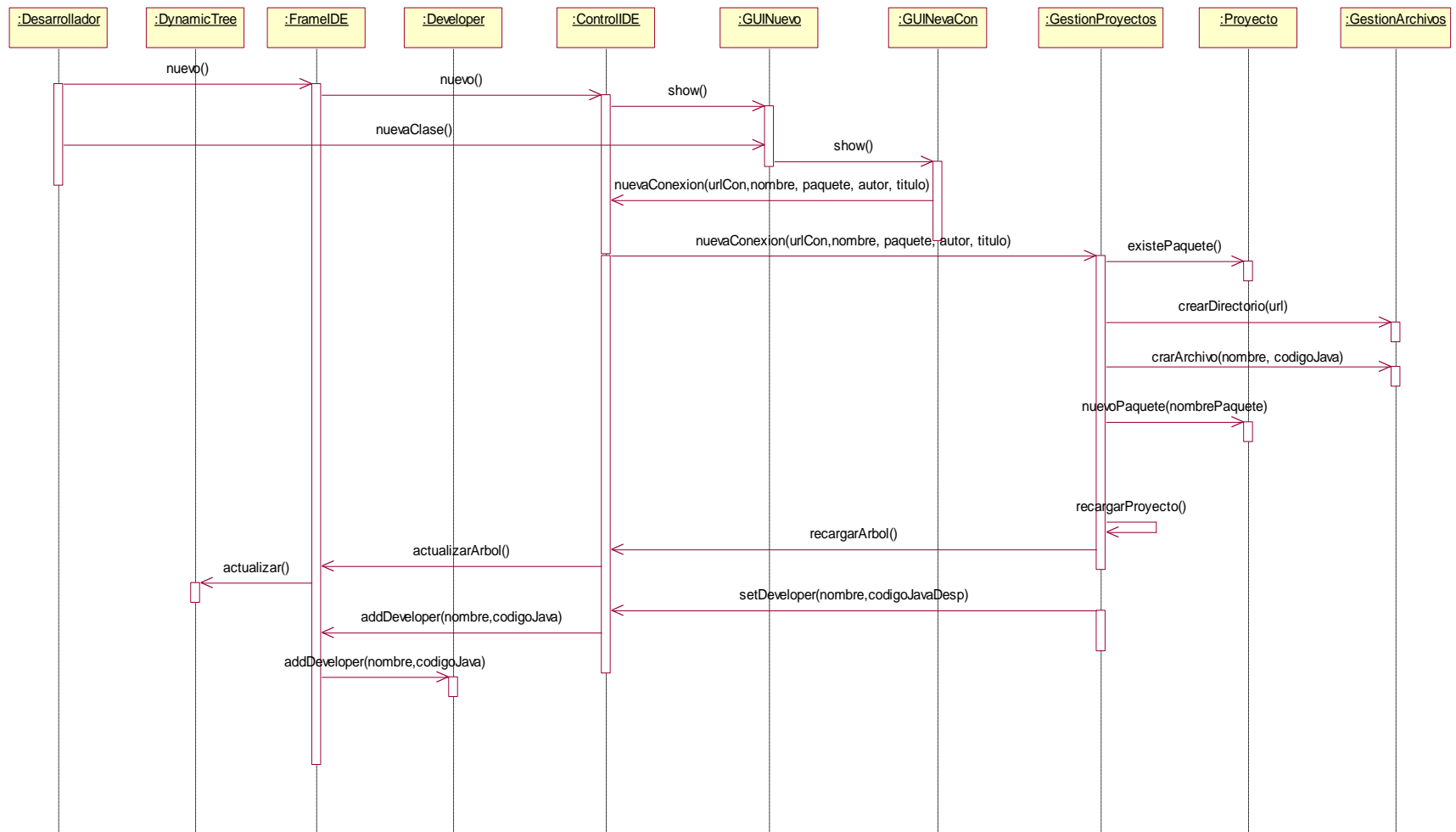
**Precondiciones:**

- Debe haber un proyecto abierto.
- El sistema debe contar con información obligatoria provista por el desarrollador (Nombre del Form y paquete).

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el icono *Nuevo* o escoge la opción *Archivo / Nuevo* en el menú principal y hace clic sobre el botón *Conexión*.
- El sistema responde mostrando una GUI Nueva Conexión
- El desarrollador suministra la información necesaria y pulsa el botón Aceptar.
- El sistema responde creando dos archivos .java, el primero una clase con los atributos y métodos necesarios para realizar la conexión HTTP y el segundo un Form con los métodos, atributos e interfaces requeridos para manejar la conexión, mostrando en el editor de código el segundo archivo, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA



**CASO DE USO No. 15: CREAR RECORD STORE**

**Actores:** Desarrollador (iniciador)

**Propósito:** Permitir al desarrollador la rápida creación de un record store para almacenamiento persistente de datos en el dispositivo Móvil

**Resumen:** El desarrollador selecciona la opción nuevo Record Store, suministra la información requerida y pulsa el botón aceptar.

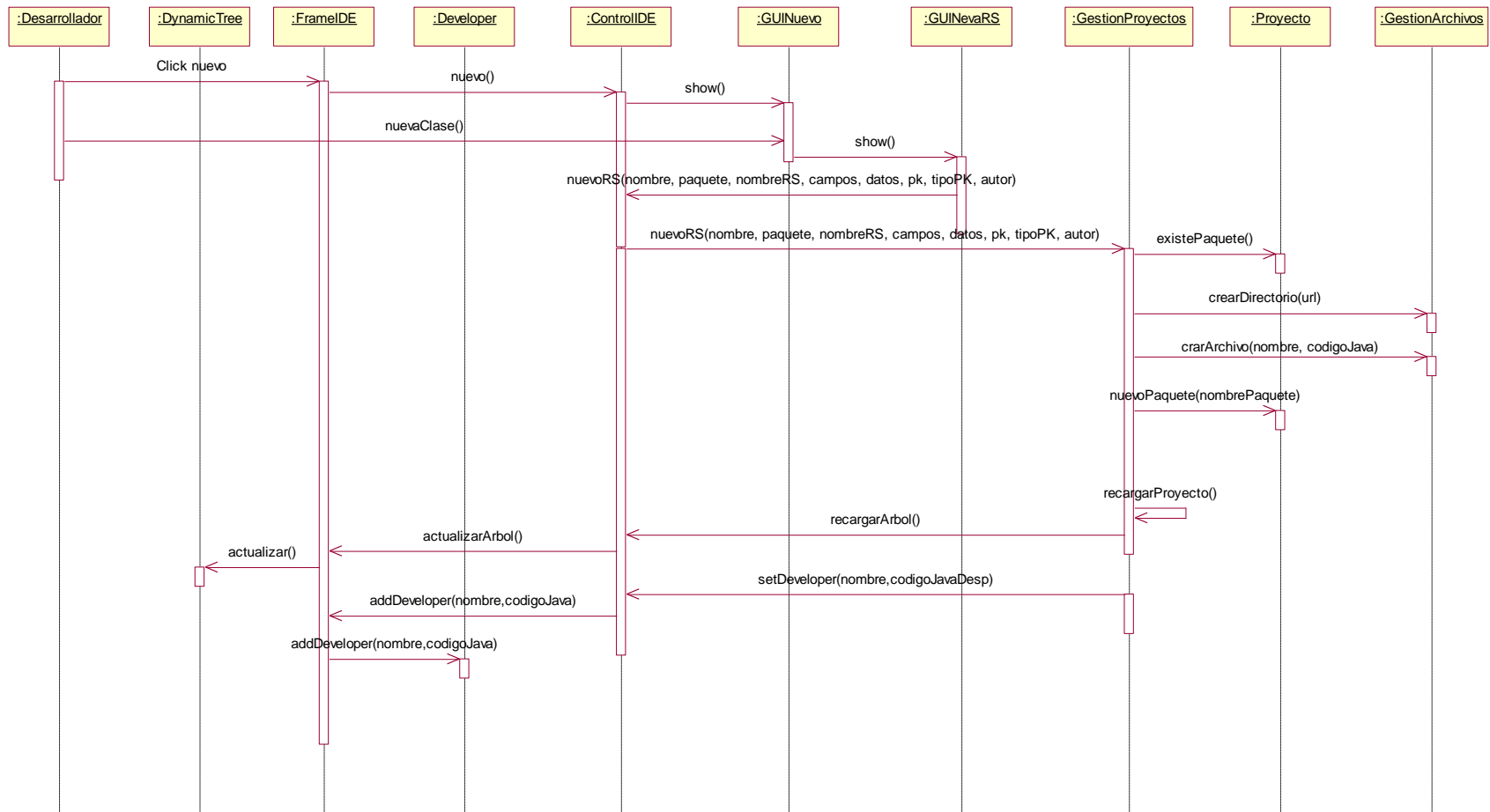
**Precondiciones:**

- Debe haber un proyecto abierto.
- El sistema debe contar con información obligatoria provista por el desarrollador (Nombre del Form y paquete).

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el icono *Nuevo* o escoge la opción *Archivo / Nuevo* en el menú principal y hace clic sobre el botón *Record Store*.
- El sistema responde mostrando una GUI Nuevo record Store
- El desarrollador suministra la información necesaria y pulsa el botón Aceptar.
- El sistema responde creando el archivo .java con el código del record store generado con los datos ingresados por el usuarios, finalizando así el caso de uso.

**DIAGRAMA DE SECUENCIA**



**CASO DE USO No. 16: CREAR GUI**

**Actores:** Desarrollador (iniciador)

**Propósito:** Permitir al desarrollador la implementación de interfaces gráficas de usuario de una manera más fácil y rápida mediante un diseñador gráfico.

**Resumen:** El desarrollador selecciona en la barra de ítems en el diseñador grafico aquel que desea adicionar a su GUI y lo arrastra hasta donde desea ubicarlo en la pantalla del emulador.

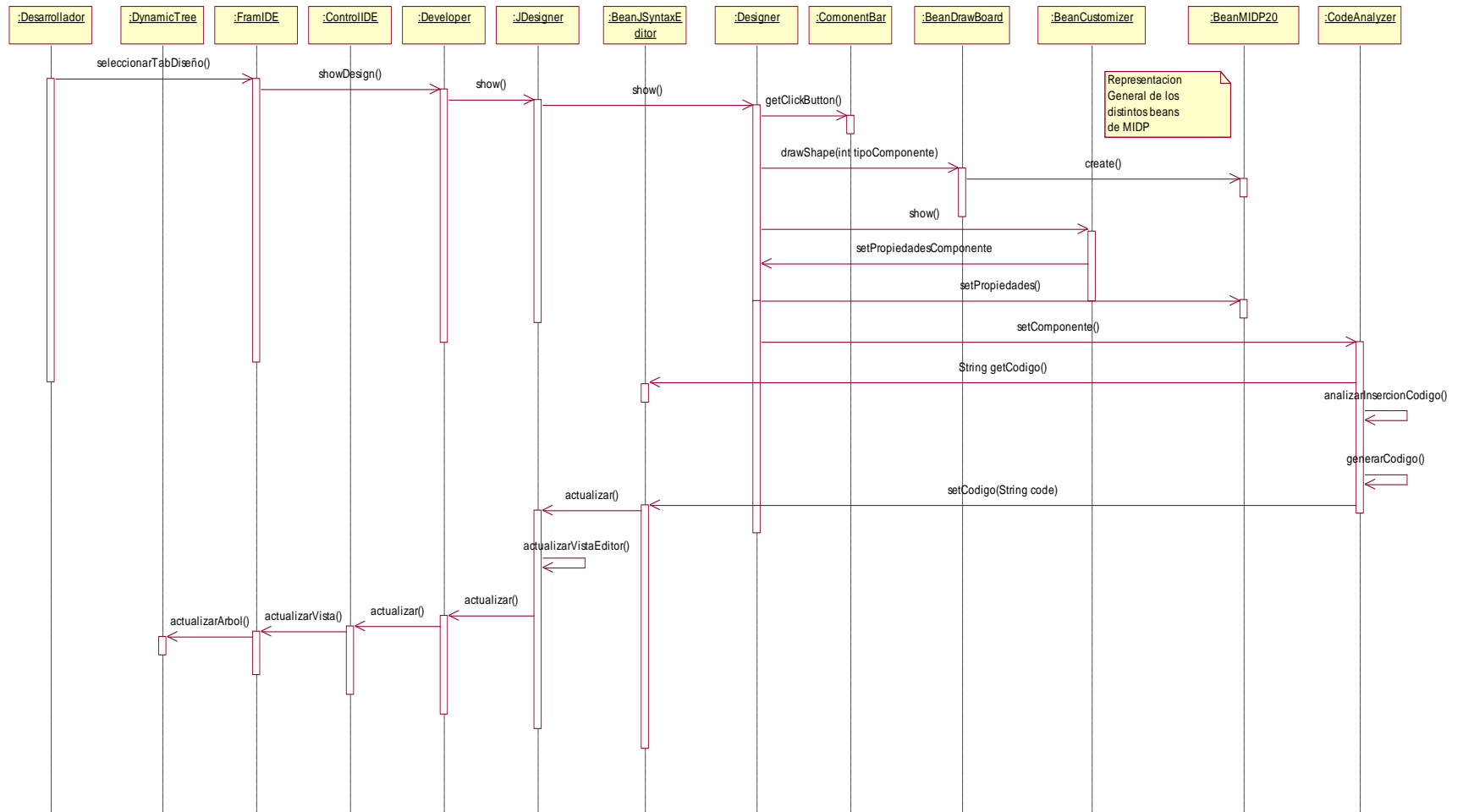
**Precondiciones:**

- Debe estar abierto un proyecto.
- Debe estar abierto el editor gráfico.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador selecciona la pestaña Diseño en la interfaz principal.
- El sistema responde mostrando una GUI de Diseño.
- El Desarrollador hace clic sobre el ítem que desea ubicar en la GUI.
- El sistema responde fijando el Java Bean que se dibujará al hacer clic sobre el área de dibujo.
- El desarrollador hace clic sobre el área de dibujo
- El sistema responde mostrando en el área de dibujo el ítem seleccionado y asigna un conjunto de propiedades por defecto al java Bean que representa el ítem dibujado
- El desarrollador hace clic sobre la pestaña fuente
- El sistema responde adicionando el código correspondiente al archivo java modificado y recargando en el editor dicho archivo, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA



**CASO DE USO No. 17: COMPILAR APLICACIÓN**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador la compilación de las aplicaciones creadas en el entorno.

**Resumen:** El desarrollador selecciona la opción compilar.

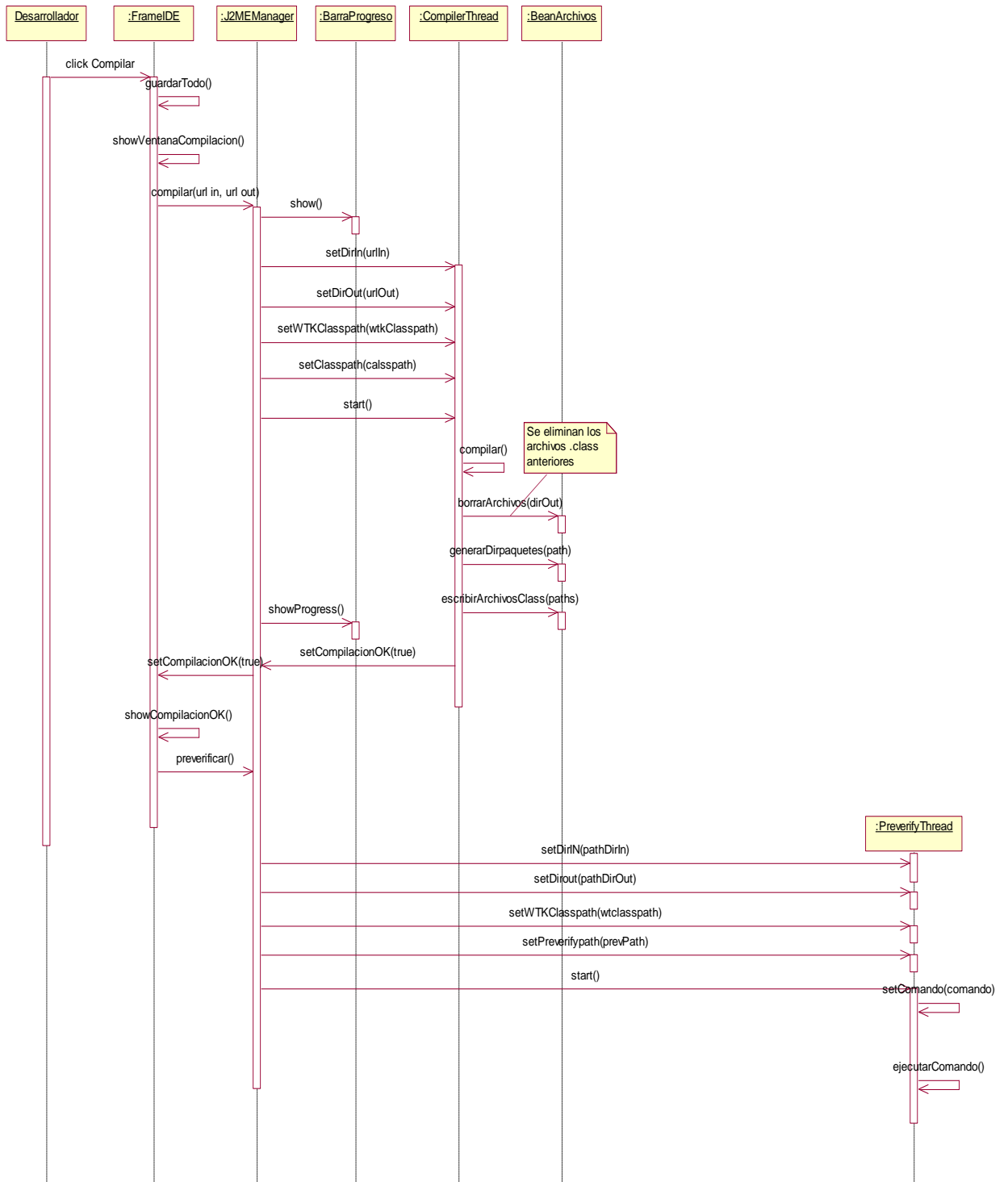
**Precondiciones:**

- Debe estar instalado un WTK versión 1.4 o posterior.
- Debe haber un proyecto abierto en el entorno.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador hace clic sobre el botón Compilar o selecciona Proyecto / Compilar proyecto en el menú principal.
- El sistema responde mostrando una barra de progreso que indica el estado de la compilación por paquetes, permitiendo visualizar el nombre del archivo que está siendo compilado.
- Si no hay errores en la compilación el sistema muestra el mensaje →Compilación completa, finalizando así el caso de uso.
- Si hay errores en la compilación el sistema muestra los errores producidos y al final el mensaje →Compilación completa, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA





**CASO DE USO No. 18: CREAR JAR**

**Actores:** Desarrollador (iniciador)

**Propósito:** Crear el archivo que contiene todos los recursos de la aplicación.

**Resumen:** el desarrollador selecciona la opción *Crear JAR* y en la GUI emergente pulsa el botón *Aceptar*.

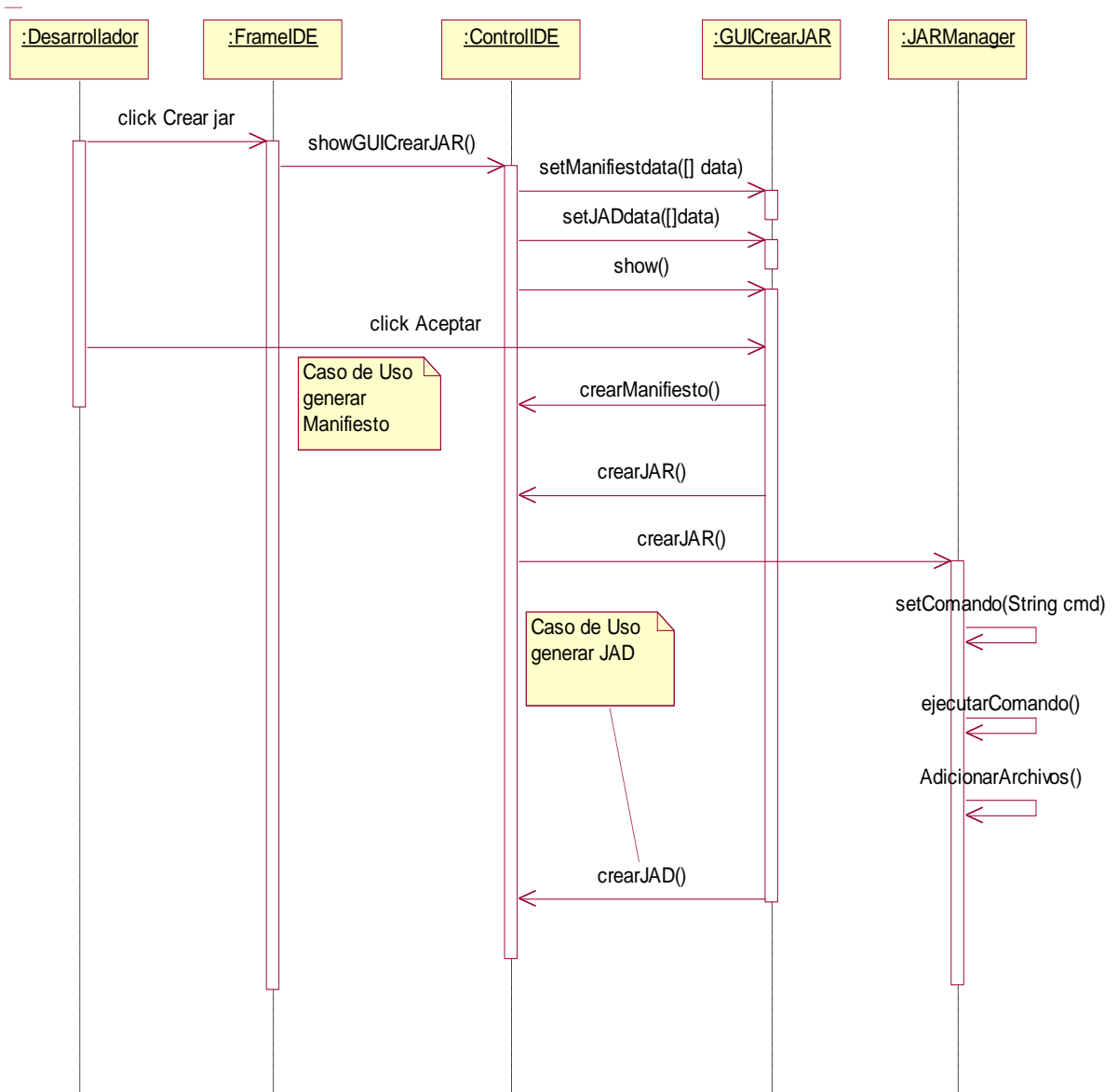
**Precondiciones:**

- No debe haber errores de compilación
- Debe haber un proyecto abierto.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador selecciona Herramientas / Crear JAR,
- El sistema responde mostrando una GUI crear Jar
- El desarrollador pulsa el botón Aceptar.
- El sistema responde llamando al caso de uso Crear Manifiesto.
- Si el manifiesto es creado correctamente, el sistema crea un archivo nombre\_proyecto.jar que contiene todas las clases (código objeto) de la aplicación desarrollada y todos los recursos (imágenes y otros archivos) que el desarrollador haya puesto en su proyecto.
- Si el archivo JAR fue creado correctamente, el sistema llama al caso de uso Crear JAD, finalizando así el caso de uso.

DIAGRAMA DE SECUENCIA



**CASO DE USO No. 19: GENERAR MANIFIESTO**

**Actores:** Caso de uso Crear JAR (iniciador).

**Propósito:** crear el archivo manifiesto que servirá para crear el archivo JAR que contendrá todos los recursos de la aplicación desarrollada.

**Resumen:** el desarrollador selecciona en el menú la opción *Crear Jar*, modifica o no el archivo manifiesto y pulsa el botón *Aceptar*.

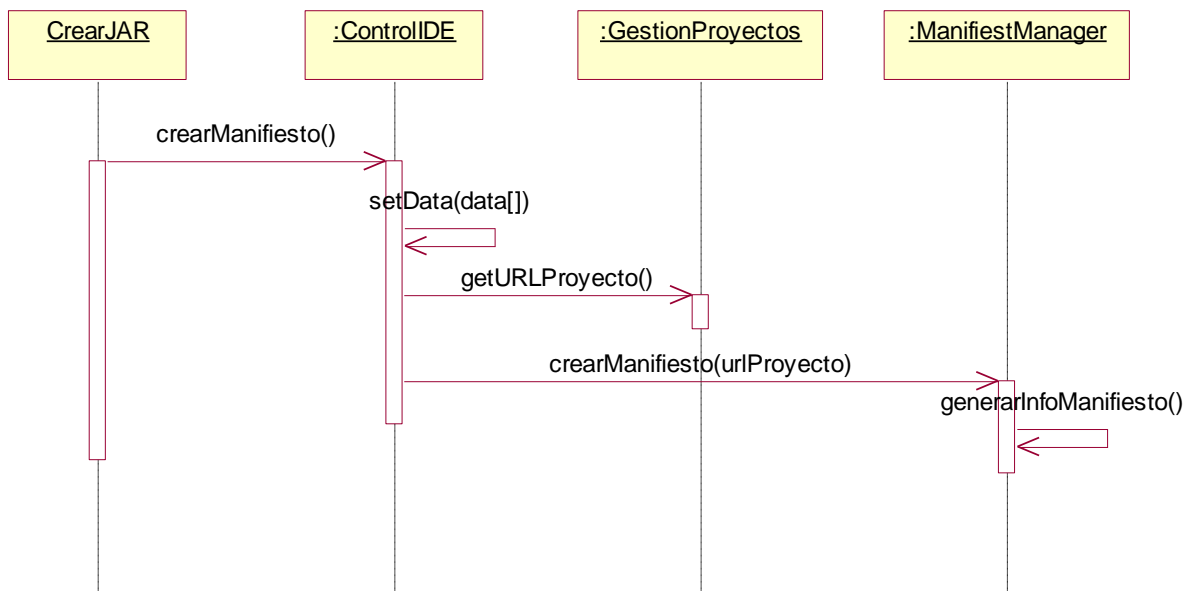
**Precondiciones:**

- No debe haber errores de compilación en la aplicación.
- Debe haber un proyecto abierto

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador selecciona en el menú Herramientas / Crear JAR.
- El sistema responde mostrando una GUI .
- Si el desarrollador modifica los valores del archivo manifiesto, el sistema responde creando un archivo de nombre MANIFIESTO.MF en la carpeta bin dentro de la carpeta del proyecto con los valores ingresados por el desarrollador finalizado así el caso de uso.
- Si el desarrollador no modifica los valores del archivo manifiesto, el sistema responde creando un archivo de nombre MANIFIESTO.MF (Figura 3.15) en la carpeta bin dentro de la carpeta del proyecto con los valores por defecto, finalizado así el caso de uso.

## DIAGRAMA DE SECUENCIA



**CASO DE USO No. 20: GENERAR JAD**

**Actores:** Caso de uso Crear JAR

**Propósito:** crear el archivo descriptor para la aplicación desarrollada.

**Resumen:** el desarrollador selecciona en el menú la opción *Crear Jar*, modifica o no el archivo JAD y pulsa el botón *Aceptar*.

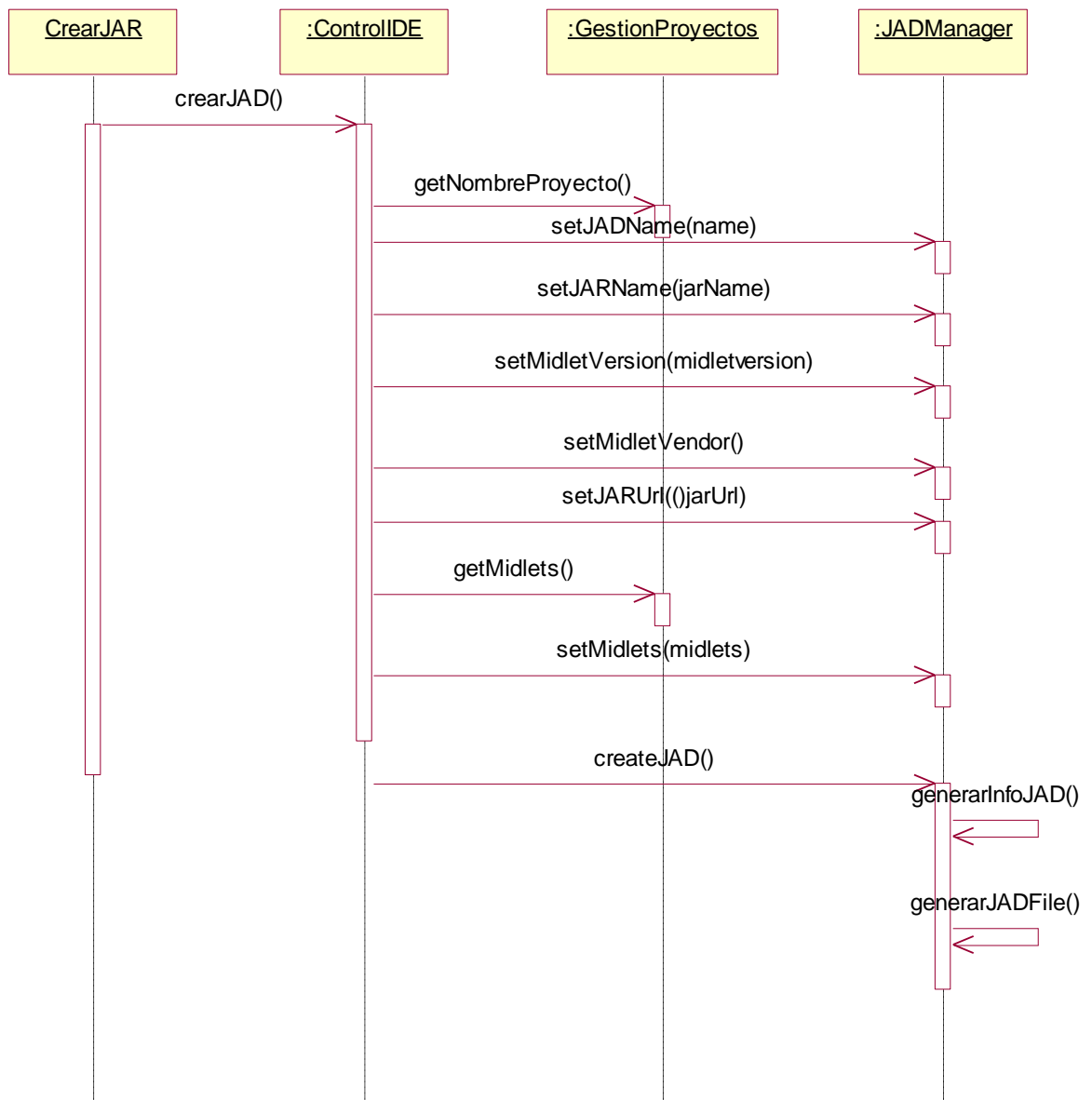
**Precondiciones:**

- Debe existir el archivo JAR.
- Debe haber un proyecto abierto.

**Flujo principal:**

- Este caso de uso se inicia cuando en el desarrollado selecciona la opción *Herramientas / Crear JAR*,
- El sistema responde mostrando una GUI.
- Si el desarrollador modifica los valores del archivo JAD, el sistema responde creando un archivo nombre\_proyecto.jad en una carpeta bin dentro de la carpeta del proyecto, con los valores ingresados por el desarrollador, finalizando así el caso de uso.
- Si el desarrollador no modifica los valores del archivo JAD, el sistema responde creando un archivo nombre\_proyecto.jad en una carpeta bin dentro de la carpeta del proyecto, con los valores por defecto, finalizando así el caso de uso.

**DIAGRAMA DE SECUENCIA**



**CASO DE USO No. 21: EMULAR APLICACIÓN**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador observar el comportamiento de la aplicación desarrollada en un emulador.

**Resumen:** El desarrollador selecciona la opción “ejecutar”

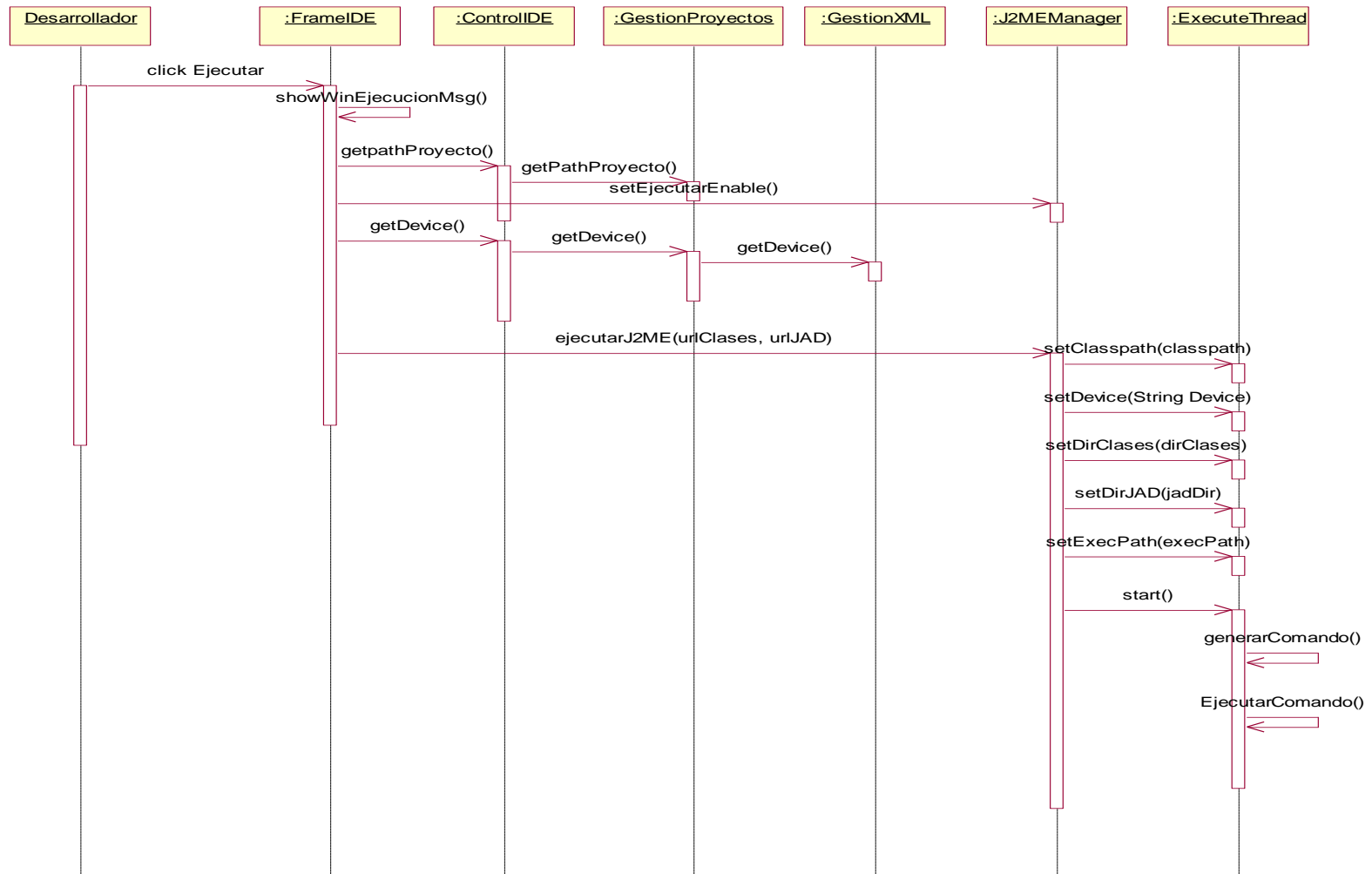
**Precondiciones:**

- La aplicación no debe presentar errores de compilación.
- Debe existir el archivo .jad.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador hace clic sobre el botón Ejecutar o escoge las opciones Ejecutar / Ejecutar en el menú principal.
- El sistema responde mostrando el emulador que haya sido configurado para el proyecto y en él, el nombre de todos los MIDlets que contenga la aplicación.
- El desarrollador selecciona y ejecuta el MIDlet deseado.
- El desarrollador cierra el emulador.
- El sistema responde mostrando las características de ejecución de la aplicación en el panel de mensajes de la plataforma, finalizando así este caso de uso.

DIAGRAMA DE SECUENCIA





**CASO DE USO No. 22: ADICIONAR BIBLIOTECAS**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador adicionar bibliotecas de clases que puedan ser utilizadas en la plataforma de desarrollo.

**Resumen:** El desarrollador selecciona la opción *Configurar Bibliotecas*, pulsa el botón *Nueva*, ingresa el nombre, selecciona los paths de la nueva bibliotecas y pulsa el botón *Aceptar* en las ventanas emergentes.

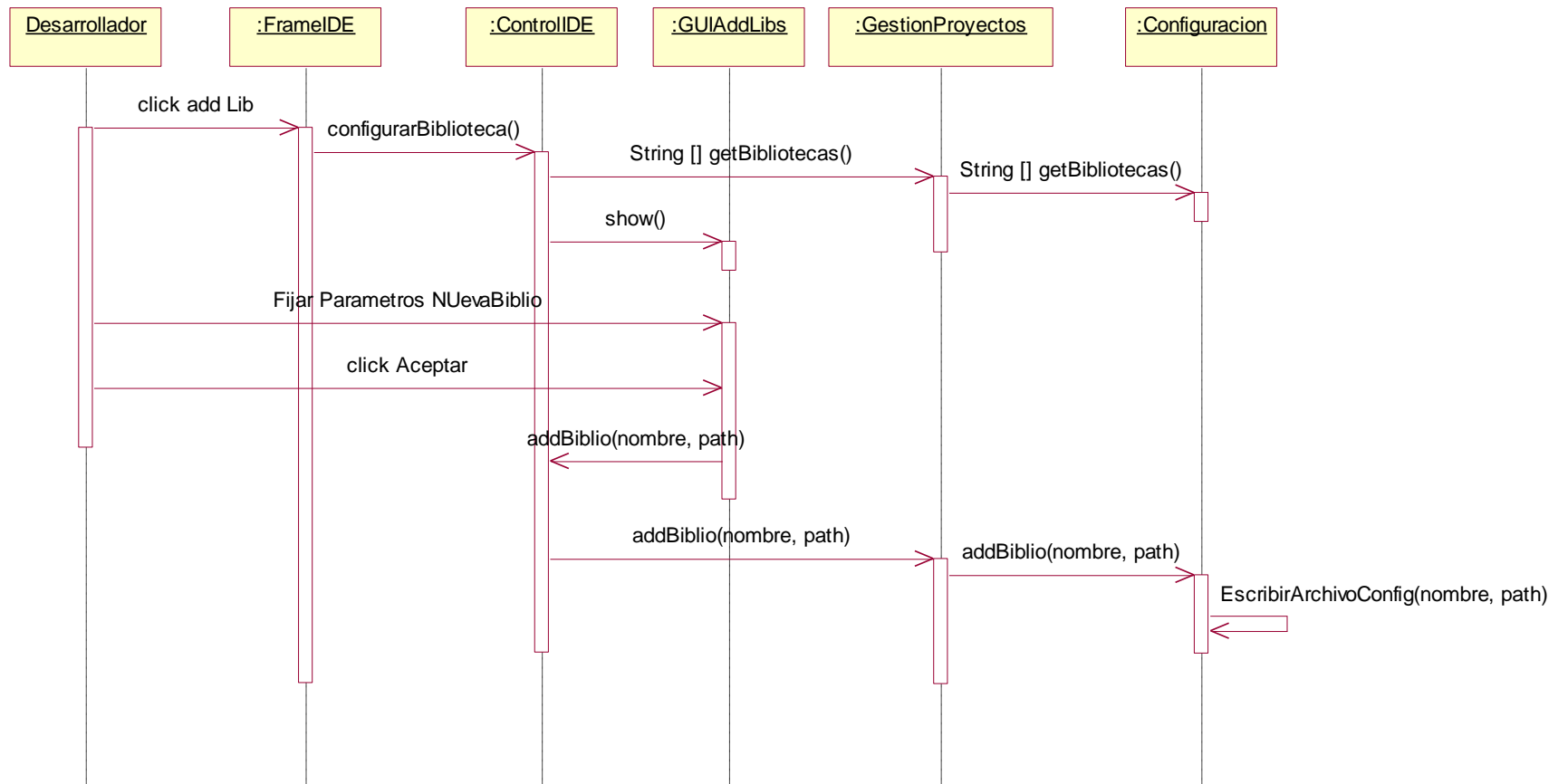
**Precondiciones:**

Ninguna.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador escoge la opción Herramientas / Configurar Bibliotecas y en la ventana emergente pulsa el botón Nueva.
- El sistema responde mostrando una ventana en donde se pide al desarrollador el nombre de la nueva biblioteca y los paths de las clases que la conformarán
- El desarrollador ingresa la información necesaria y pulsa la el botón Aceptar.
- El sistema responde mostrando las bibliotecas configuradas en el entorno.
- El desarrollador pulsa el botón Aceptar.
- El sistema responde grabando en el archivo de configuración de la plataforma el nombre de la nueva biblioteca y los paths de las clases que la componen, finalizando así este caso de uso.

**DIAGRAMA DE SECUENCIA**



**CASO DE USO No. 23: MODIFICAR BIBLIOTECAS**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador modificar los paths de las bibliotecas de clases que estén configuradas en la plataforma de desarrollo.

**Resumen:** el desarrollador selecciona la opción *Herramientas / Configurar Bibliotecas*, selecciona la biblioteca que desea modificar, adiciona o elimina rutas a esa biblioteca y pulsa *Aceptar*.

**Precondiciones:**

Deben existir bibliotecas configuradas en la plataforma de desarrollo.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador escoge la opción *Herramientas / Configurar Bibliotecas*.
- El sistema responde mostrando una ventana.
- El desarrollador selecciona en el árbol de bibliotecas aquella que desea modificar.
- El sistema responde mostrando el nombre de la biblioteca y las rutas de las clases que la conforman.
- Si el desarrollador pulsa el botón Adicionar el sistema responde mostrando una ventana en la que el desarrollador puede seleccionar la nueva ruta que hará parte de la biblioteca.
- Si el desarrollador selecciona una de las rutas que conforman la biblioteca y pulsa el botón Remove el sistema responde borrando de la interfaz gráfica la ruta seleccionada.
- El desarrollador pulsa el botón Aceptar.
- El sistema responde guardando en el archivo de configuración de la plataforma (archivo config.xml) los cambios hechos a la biblioteca seleccionada, finalizando así este caso de uso.

**CASO DE USO No. 24: ELIMINAR BIBLIOTECAS**

**Actores:** Desarrollador (iniciador)

**Propósito:** permitir al desarrollador eliminar las bibliotecas de clases que haya adicionado a la plataforma de desarrollo.

**Resumen:** El desarrollador elige la opción *Herramientas / Configurar Bibliotecas*, selecciona en el árbol de bibliotecas aquella que desee eliminar y pulsa el botón *Eliminar*.

**Precondiciones:**

Deben existir bibliotecas configuradas en la plataforma de desarrollo.

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador escoge la opción *Herramientas / Configurar Bibliotecas*.
- El sistema responde mostrando una ventana.
- El desarrollador selecciona en el árbol de bibliotecas aquella que desea eliminar y pulsa el botón *Eliminar*.
- El sistema responde borrando la biblioteca seleccionada del archivo de configuración de la plataforma, finalizando así este caso de uso.

**CASO DE USO No. 25: ADICIONAR EMULADOR**

**Actores:** Desarrollador (iniciador).

**propósito:** permitir al desarrollador la utilización no solo de los emuladores que vienen con el WTK de Sun Microsystems, sino también la utilización de emuladores de otras empresas que sean compatibles con este.

**Resumen:** El desarrollador ubica el nuevo emulador en el directorio *wtklib / devices* del WTK utilizado.

**Precondiciones:**

El nuevo emulador debe ser compatible con el WTK utilizado.

**Flujo principal:**

El desarrollador ubica el nuevo emulador en el directorio *wtklib / devices* del WTK utilizado.

**CASO DE USO No. 26: SELECCIONAR EMULADOR**

**Actores:** Desarrollador(iniciador).

**Propósito:** permitir al desarrollador utilizar cualquiera de los emuladores disponibles en el WTK utilizado, para observar la ejecución de su aplicación.

**Resumen:** El desarrollador selecciona en el menú la opción *Herramientas / Configurar Emulador*, selecciona el emulador que desea utilizar y pulsa el botón *Aceptar*.

**Precondiciones:**

- Debe haber un proyecto abierto en el entorno.
- Debe haber un WTK 1.4 o posterior instalado

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador selecciona en el menú la opción herramientas / Configurar Emulador.
- El sistema responde mostrando una GUI .
- El desarrollador selecciona de la lista de emuladores aquel que desea utilizar y pulsa el botón Aceptar.
- El sistema responde guardando en el archivo de configuración del proyecto (figura 3.6) el dispositivo que se utilizará como emulador, finalizando de esta forma el caso de uso.

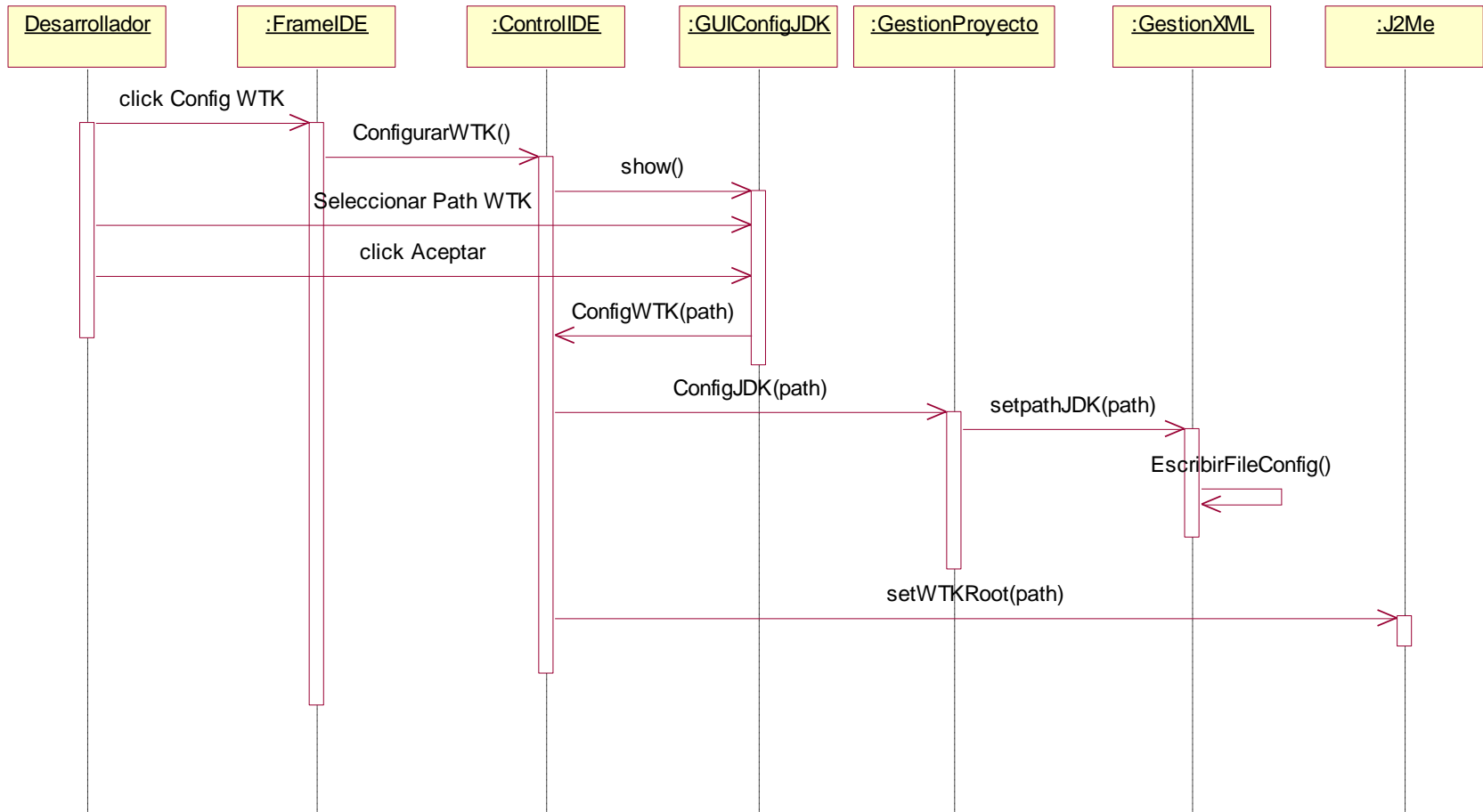
**CASO DE USO No. 27: CONFIGURAR WTK****Actores:** Desarrollador (iniciador)**Propósito:** permitir al desarrollador escoger el WTK que desea utilizar en su aplicación de aquellos que tenga instalados en su sistema.**Resumen:** el desarrollador elige la opción *Herramientas / Configurar WTK*, ingresa la ruta del directorio raíz del WTK que desea utilizar y pulsa *Aceptar*.**Precondiciones:**

- Debe haber un proyecto abierto en el entorno.
- Debe haber un WTK 1.4 o posterior instalado

**Flujo principal:**

- Este caso de uso inicia cuando el desarrollador selecciona en el menú principal la opción *Herramientas / Configurar WTK*.
- El sistema responde desplegando una ventana en la que se le solicita al desarrollador la ruta del directorio raíz del WTK que desea utilizar para su aplicación .
- El usuario ingresa la ruta del WTK y pulsa la el botón *Aceptar*.
- El sistema responde guardando en el archivo de configuración del proyecto la ruta del WTK configurado, finalizando así el caso de uso.

### DIAGRAMA DE SECUENCIA





**CASO DE USO No. 28: OBTENER AYUDA DEL ENTORNO**

**Actores:** Desarrollador

**Propósito:** Permitir que el desarrollador obtenga ayuda acerca de cómo realizar alguna operación dentro de la plataforma de desarrollo.

**Resumen:** el desarrollador elige la opción *Ayuda / Ayuda del IDE* y en la ventana emergente selecciona el tema relacionado con la operación que desea realizar.

**Precondiciones:**

Ninguna.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador selecciona en el menú la opción *Ayuda / Ayuda del IDE*.
- El sistema responde mostrando una GUI de ayuda.
- El desarrollador busca los temas relacionados con la ayuda que necesita.
- El desarrollador cierra la ventana finalizando así el caso de uso.

**CASO DE USO No. 29: OBTENER AYUDA DE J2ME**

**Propósito:** Permitir que el desarrollador obtenga ayuda acerca de las APIs de J2ME.

**Resumen:** el desarrollador elige la opción *Ayuda / Ayuda de J2ME* y en la ventana emergente selecciona el tema relacionado con la ayuda que necesita.

**Precondiciones:**

Debe haber un WTK instalado.

**Flujo principal:**

- Este caso de uso se inicia cuando el desarrollador selecciona en el menú la opción *Ayuda / Ayuda del IDE*.
- El sistema responde abriendo en un explorador (Internet Explorer para Windows o Mozilla para Linux) y desplegando la pagina *Index.html* de la documentación del WTK utilizado, finalizando así el caso de uso.

**DIAGRAMA DE CLASES**

Ver en el CD. Diagrama de clases