

Instituto Tecnológico de Costa Rica
Ingeniería en Computación
Programación Orientada a Objetos
I Semestre 2015
Mauricio Avilés

Proyecto #2

Estudiantes:
Kenneth Callow
Sara Castro Saenz

Índice

Introducción	4
Casos de uso	5
Diagramas de actividad	12
Modelo Conceptual	15
Diagramas de clase	16
Diagrama de secuencia	17
Conclusiones y recomendaciones	18
Referencias	19

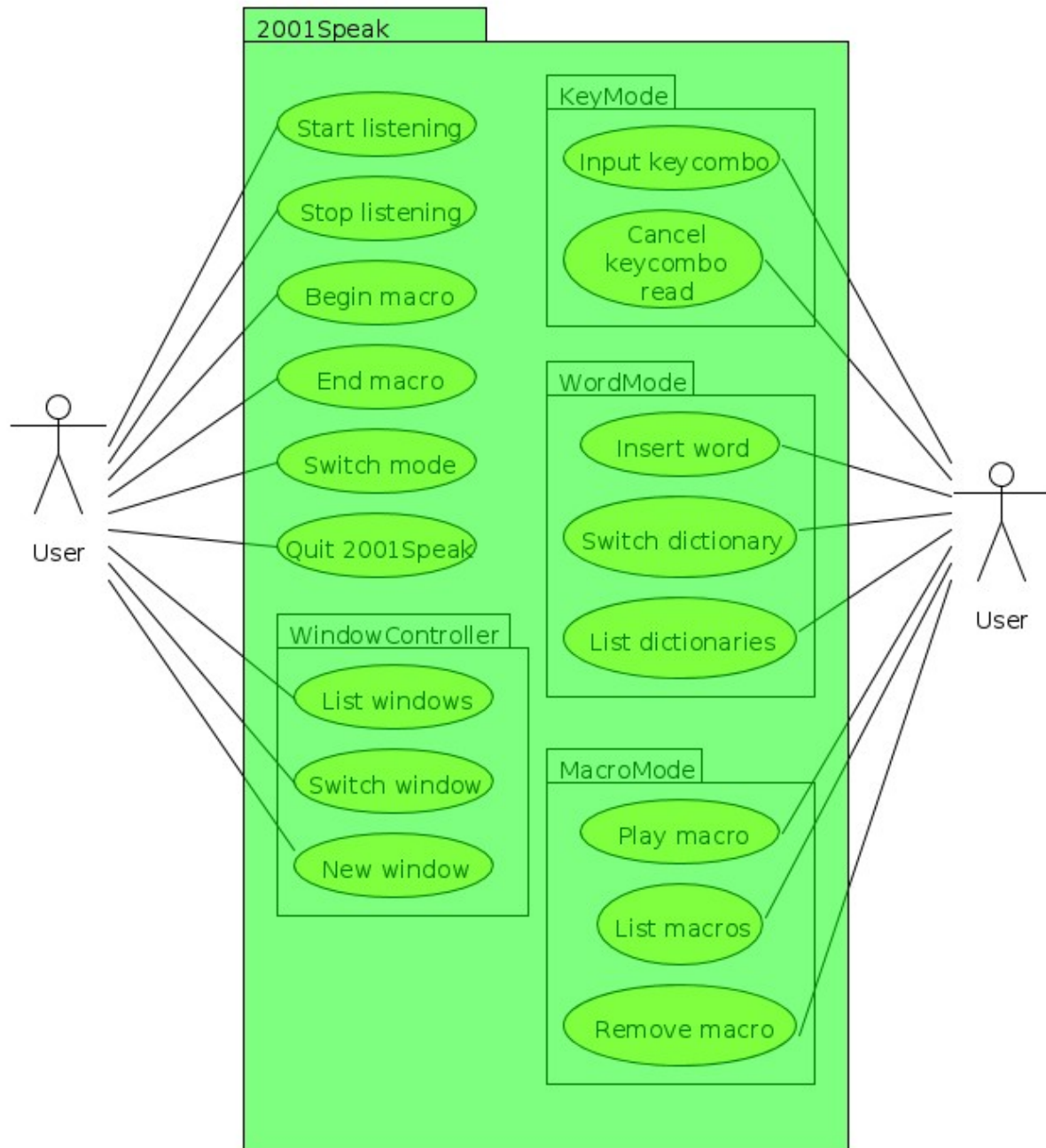
Introducción

La ciencia ficción nos pinta un mundo donde los humanos hablan con máquinas con características profundamente humanas, a tal punto que nos superan en el intelecto y existen conflictos de poder entre nosotros. Este proyecto, titulado 2001Speak, busca llevar estas fantasías un paso más cerca a la realidad, dados los importantes avances en la tecnología de reconocimiento de voz, y la falta de herramientas de propósito general para comunicarse con las computadoras.

2001Speak consiste en un administrador de ventanas de Linux implementada en C++ que utiliza la librería en C PocketSphinx de Carnegie Mellon University, la cual interpreta las entradas audibles desde el micrófono del usuario a texto. 2001Speak interpreta estas entradas de texto como comandos a insertar y ejecutar en la terminal. implementa la posibilidad de definir comandos personalizados por el usuario, los cuales consisten en palabras clave que se expanden en comandos más largos, secuencias de comandos, o en combinaciones de teclas.

Por ejemplo, las palabras “bright 250” podrían convertirse en “echo 250 | tee /sys/class/backlight/intel_backlight/brightness”, lo cual pone el brillo en 25%; “close” podría convertirse en la combinación Alt-F4. Los comandos personalizados se crean en tiempo de ejecución, o sea, por medio de comandos de audio hablados por el usuario para ese propósito, por lo que le abre las puertas a todos los usuarios, aún poco técnicos, de “programar” sus máquinas con sólo hablar. También se implementan los macros, los cuales consisten en “grabaciones” de todas las acciones que realiza el usuario, que son guardadas en un script de shell para cada una dentro de la carpeta de configuración de 2001Speak. El usuario indica cuándo comenzar a grabar y cuándo terminar, y un comando al cual asignarle el script. Esta herramienta, combinada con las anteriores, crea todo un sistema sobre el cual el usuario puede controlar su máquina, de una forma mucho más poderosa y mucho más amistosa que el uso normal de las interfaces de texto en Linux e interfaces en general. También es útil para introducir conceptos de programación, ya que los hace conceptualmente más accesibles y tangibles a los no programadores.

Casos de Uso



Descripción de los casos de uso

Nombre	Start Listening
Objetivo	Escuchar los comandos que el usuario indique
Precondiciones	Tener abierto el programa
Condición de terminación exitosa	Los comandos se procesan con éxito
Condición de terminación fallida	Falla procesar los comandos
Desencadenador	El usuario activa la escucha presionando el botón o abriendo el programa.
Flujo principal	1. Escuchar los comandos 2. Procesar los comandos
Extensiones	

Nombre	Stop Listening
Objetivo	Detener el proceso de listening
Precondiciones	Start Listening
Condición de terminación exitosa	El proceso se detiene
Condición de terminación fallida	El proceso no se detiene
Desencadenador	El usuario indica por medio de la voz que desea desactivar la escucha
Flujo principal	1. El usuario indica por medio de comandos de voz que desea detener la escucha
Extensiones	

Nombre	Play Macro
Objetivo	Ejecutar uno de los macros que esta guardado en el directorio de macros
Precondiciones	Debe estar guardado en el directorio de macros
Condición de terminación exitosa	Los macros se ejecutan
Condición de terminación fallida	No puede procesar los macros
Desencadenador	El usuario indica por medio de la voz o el shell el comando para iniciar el macro que desea
Flujo principal	1. El usuario indica que desea ejecutar macros 2. Play Macro se activa para escuchar al usuario
Extensiones	

Nombre	End Macro
Objetivo	Terminar el proceso de Play Macro
Precondiciones	Play Macro
Condición de terminación exitosa	El proceso termina
Condición de terminación fallida	Falla la detención del proceso
Desencadenador	El usuario indica que quiere terminar el proceso por medio de la voz
Flujo principal	1. El usuario indica que desea terminar 2. Termina el proceso de Play Macro
Extensiones	

Nombre	Switch Mode
Objetivo	Cambiar el modo
Precondiciones	ninguna
Condición de terminación exitosa	Se cambia al modo que el usuario solicita
Condición de terminación fallida	Falla el cambio
Desencadenador	El usuario indica a que modo desea cambiar por medio de la voz
Flujo principal	1. El usuario indica con la voz a que modo desea cambiar 2. El programa cambia de modo
Extensiones	

Nombre	Quit 2001Speak
Objetivo	Cerrar el programa
Precondiciones	Tener el programa abierto
Condición de terminación exitosa	El programa se cierra
Condición de terminación fallida	El programa no se cierra
Desencadenador	El usuario indica por medio de la voz que desea de cerrar el programa
Flujo principal	1. El usuario indica que desea cerrar el programa usando comandos de voz
Extensiones	

Nombre	List Windows
Objetivo	Mostrar una lista de las ventanas que están ligadas al programa
Precondiciones	Tener al menos una ventana ligada
Condición de terminación exitosa	Se visualiza la lista
Condición de terminación fallida	No se muestra la lista
Desencadenador	El usuario indica por medio de la voz que sea ver la lista de ventanas
Flujo principal	1. El usuario indica por medio de los comandos que desea ver la lista 2. Se visualiza la lista
Extensiones	2.1. Se comprueba que hay al menos una ventana abierta

Nombre	Switch Windows
Objetivo	Cambiar las ventanas
Precondiciones	Al menos debe de haber 2 ventanas
Condición de terminación exitosa	La ventana cambia
Condición de terminación fallida	No cambian las ventanas
Desencadenador	El usuario indica por medio de un comando de voz que desea cambiar las ventanas
Flujo principal	1. El usuario indica que desea cambiar las ventanas 2. La ventana cambia
Extensiones	2. Comprueba que hay al menos 2 ventanas

Nombre	New Window
Objetivo	Abrir una nueva ventana
Precondiciones	Ninguna
Condición de terminación exitosa	La ventana se abre
Condición de terminación fallida	Falla al abrir la ventana
Desencadenador	El usuario indica por medio de la voz que desea abrir una ventana
Flujo principal	1. El usuario indica que desea abrir la ventana 2. La ventana se abre
Extensiones	

Nombre	Input Keycombo
Objetivo	Ingresar un nuevo combo del teclado
Precondiciones	El combo no debe de existir
Condición de terminación exitosa	El combo se guarda
Condición de terminación fallida	El combo no se guarda
Desencadenador	El usuario indica por medio de un comando de voz que desea ingresar un nuevo combo
Flujo principal	1. El usuario indica que desea ingresar un nuevo combo 2. Ingresa el combo cuando el programa le indica 3. Se guarda
Extensiones	3.1. Comprueba que ya no este en la lista de los combos

Nombre	Cancel Keycombo Read
Objetivo	Cancelar la lectura el combo
Precondiciones	Haber empezado a grabar un combo
Condición de terminación exitosa	Se cancela la lectura y no lo guarda
Condición de terminación fallida	Falla la cancelación
Desencadenador	El usuario indica por medio de la voz que desea cancelar
Flujo principal	1. El usuario indica que desea agregar un nuevo combo 2. El programa le solicita el nuevo combo 3. El usuario cancela
Extensiones	

Nombre	Insert Word
Objetivos	Escribir las palabras que el usuario indica por medio de la voz
Precondiciones	Estar en <i>Word Mode</i>
Condición de terminación exitosa	Las palabras se procesan exitosamente
Condición de terminación fallida	Falla procesar las palabras
Desencadenador	El usuario indica que desea escribir lo que dice
Flujo principal	1. Entrar a <i>Word Mode</i> 2. El usuario indica por medio de comandos de voz que desea escribir
Extensiones	

Nombre	List Dictionaries
Objetivo	Ver la lista de diccionarios
Precondiciones	Estar en <i>Word Mode</i>
Condición de terminación exitosa	La lista se muestra
Condición de terminación fallida	Falla la visualización de la lista
Desencadenador	El usuario indica por medio de la voz que desea ver la lista de diccionarios
Flujo principal	<ol style="list-style-type: none"> 1. Entrar a <i>Word Mode</i> 2. Solicitar ver la lista por comandos de voz 3. Se muestra la lista
Extensiones	

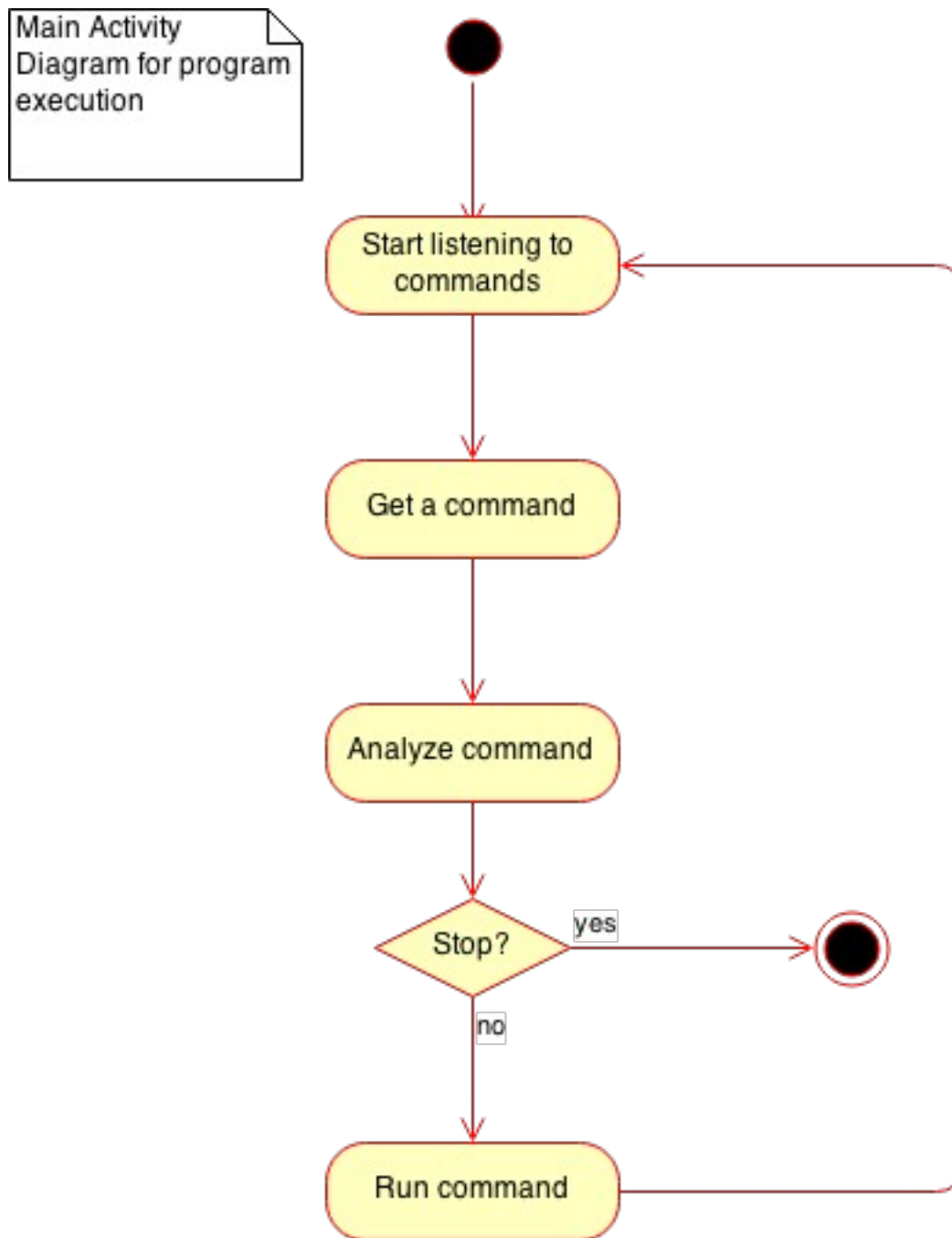
Nombre	Switch Dictionary
Objetivo	Cambiar diccionarios entra comandos Unix, texto corriente y otros lenguajes
Precondiciones	Estar en <i>Word Mode</i>
Condición de terminación exitosa	El diccionario se cambia
Condición de terminación fallida	El diccionario no se cambia
Desencadenador	El usuario indica por medio de la voz que desea cambiar los diccionarios
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a <i>Word Mode</i> 2. El usuario solicita cambiar de diccionario por medio de la voz 3. Cambia el diccionario
Extensiones	

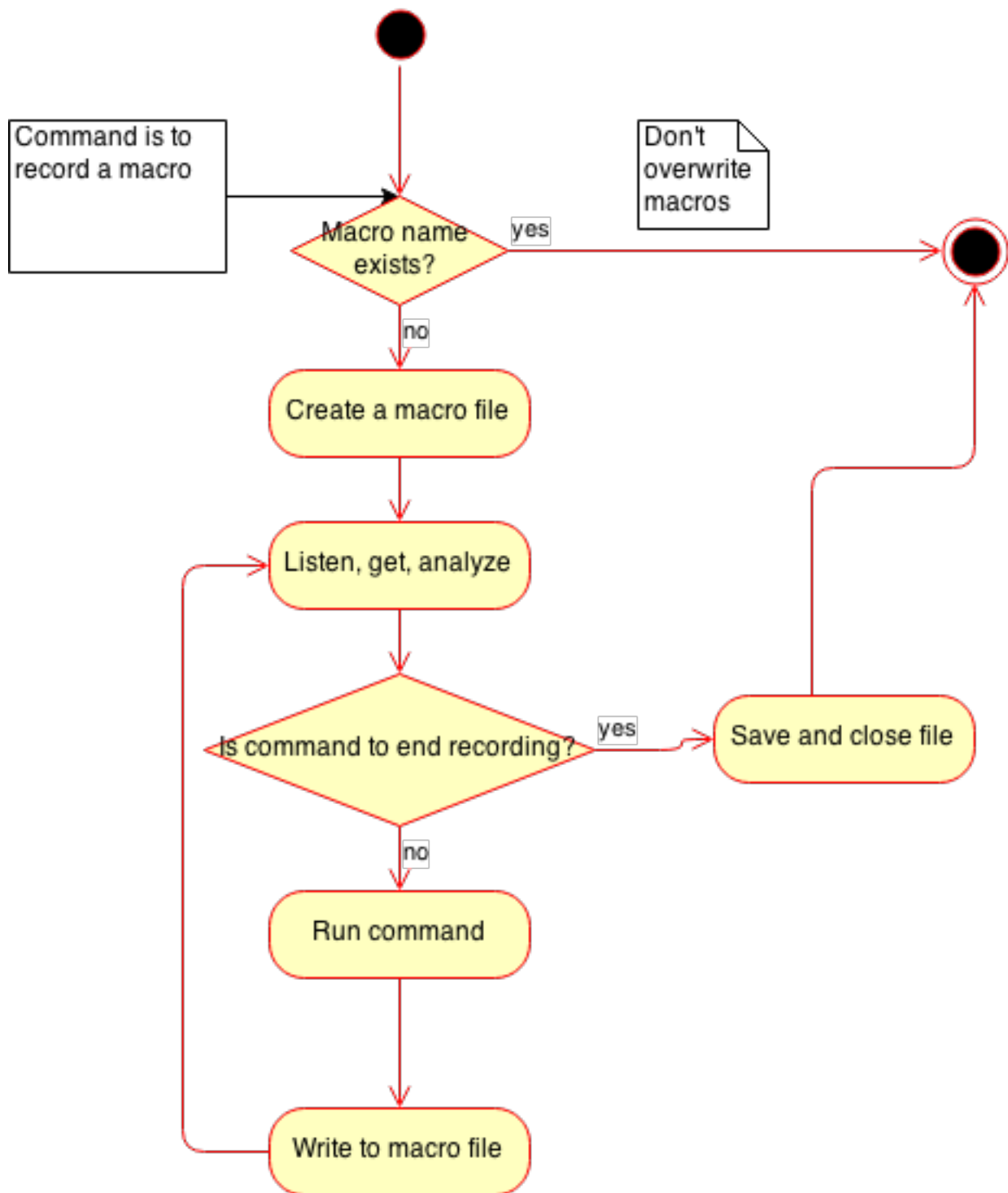
Nombre	Begin Macro
Objetivo	Grabar macro
Precondiciones	Que no exista
Condición de terminación exitosa	El macro se graba con éxito
Condición de terminación fallida	Falla la grabación del macro
Desencadenador	El usuario indica por medio de la voz cuando desee iniciar a grabar el macro
Flujo principal	<ol style="list-style-type: none"> 1. El usuario indica que desea grabar
Extensiones	

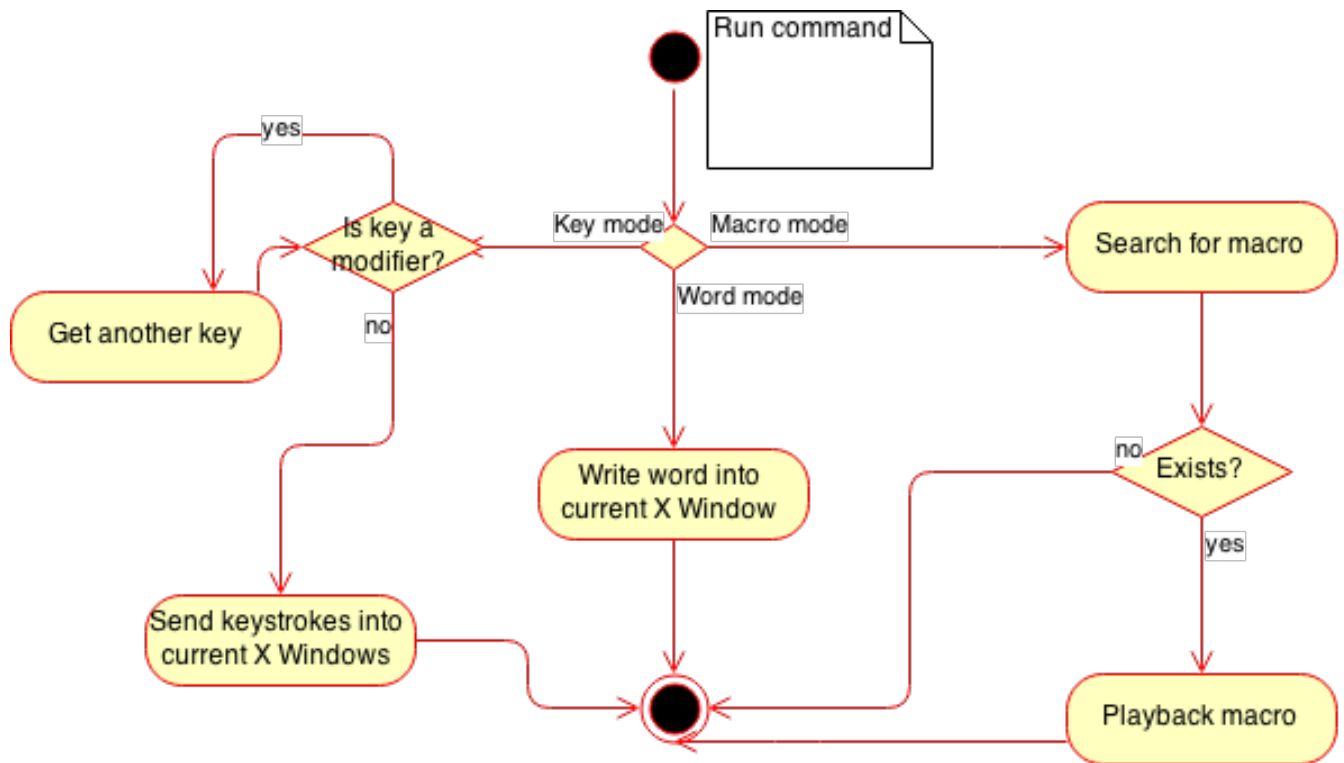
Nombre	List Macros
Objetivo	Mostrar la lista de macros
Precondiciones	Estar en <i>Macro Mode</i>
Condición de terminación exitosa	La lista se muestra
Condición de terminación fallida	La lista no se muestra
Desencadenador	El usuario indica por medio de la voz que desea ver la lista de macros
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a <i>Macro Mode</i> 2. El usuario el usuario solicita ver la lista 3. La lista se muestra
Extensiones	

Nombre	Remove Macro
Objetivo	Eliminar un macro
Precondiciones	El macro debe existir
Condición de terminación exitosa	El macro se elimina con éxito
Condición de terminación fallida	Falla el borrado del macro
Desencadenador	El usuario presiona el botón para borrar el macro
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa a <i>Macro Mode</i> 2. El usuario presiona el botón para eliminar un macro 3. El macro se elimina
Extensiones	<ol style="list-style-type: none"> 2.1. Comprueba que el macro existe

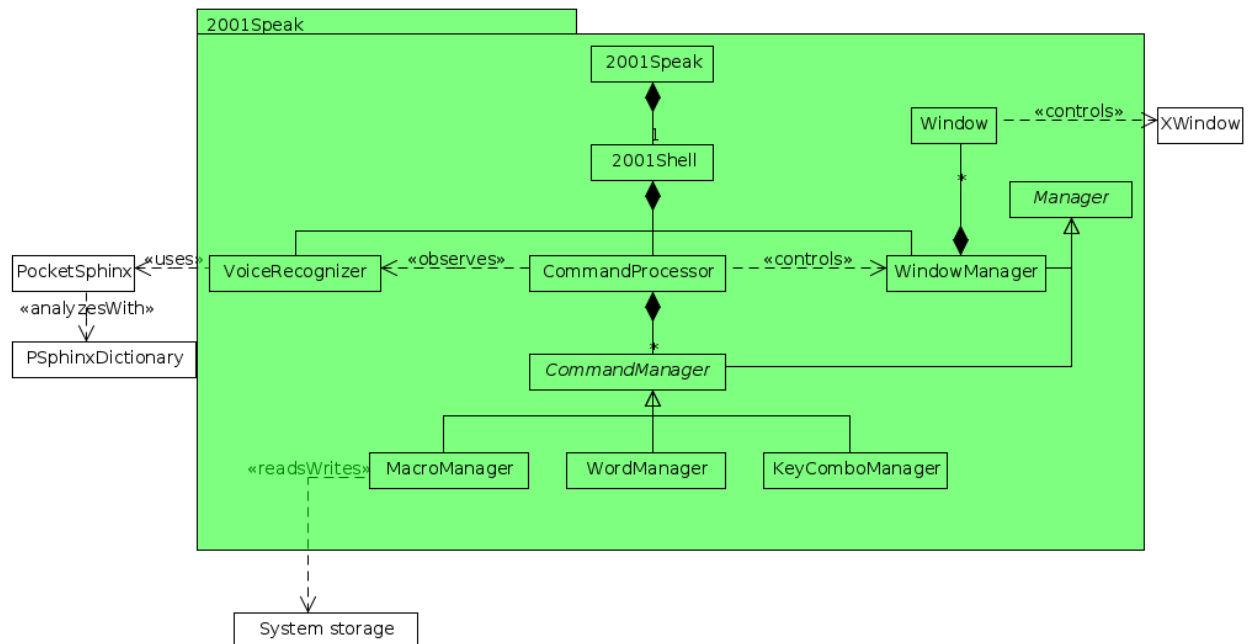
Diagramas de Actividad



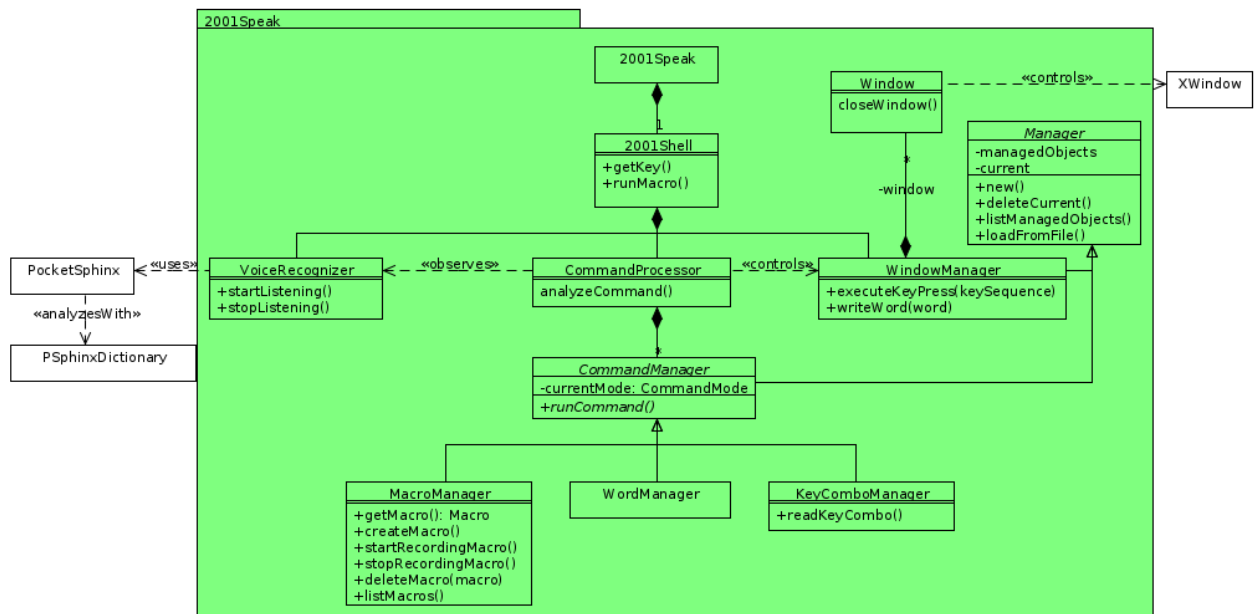




Modelo Conceptual

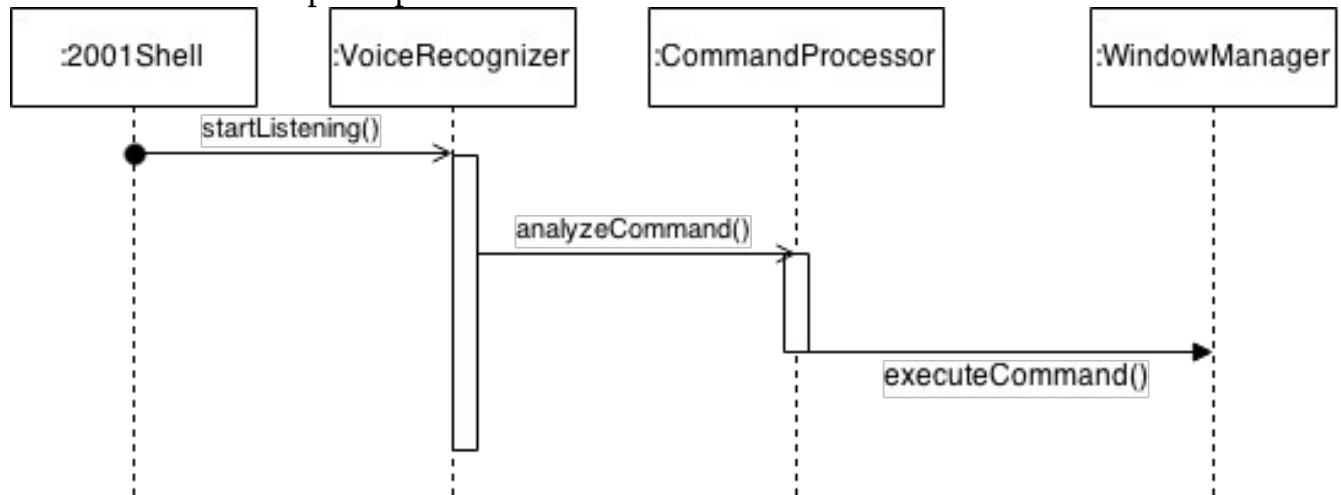


Diagramas de Clase

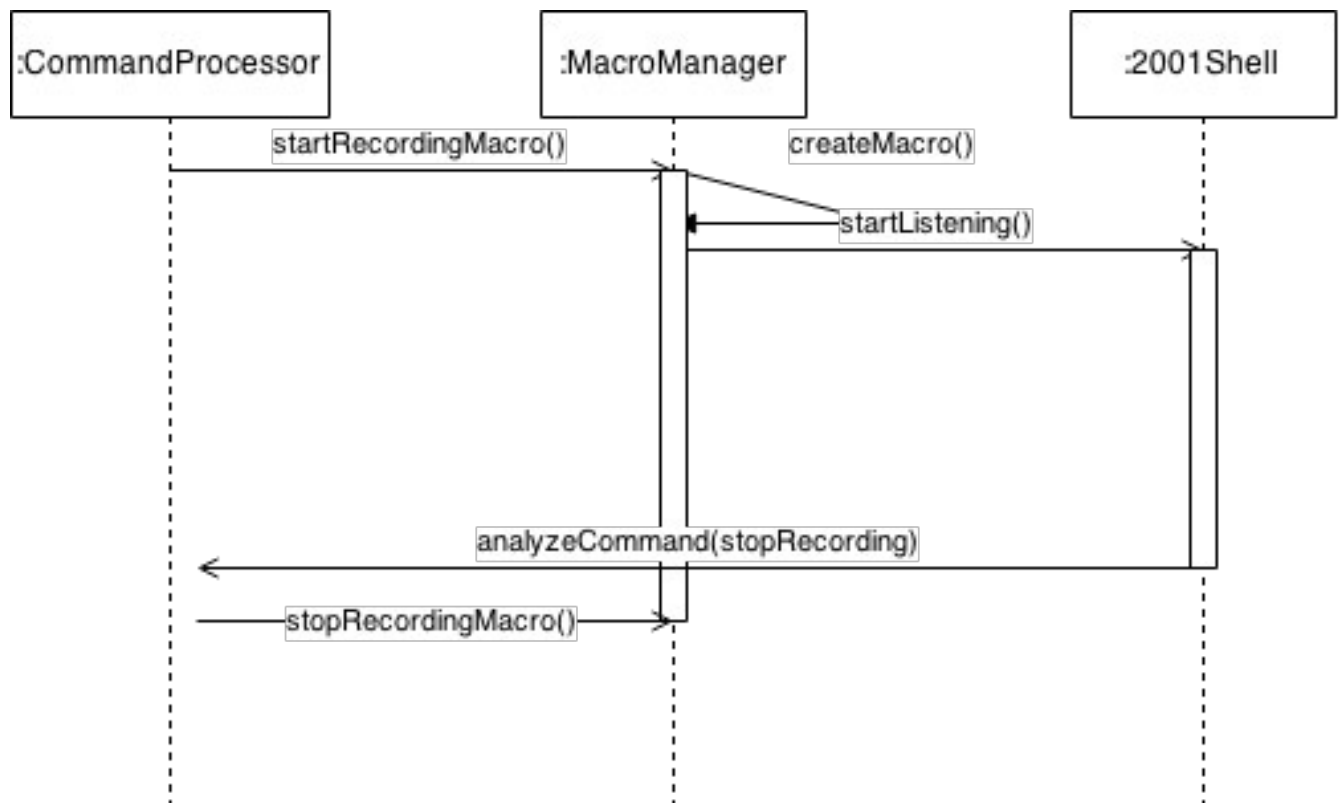


Diagramas de Secuencia

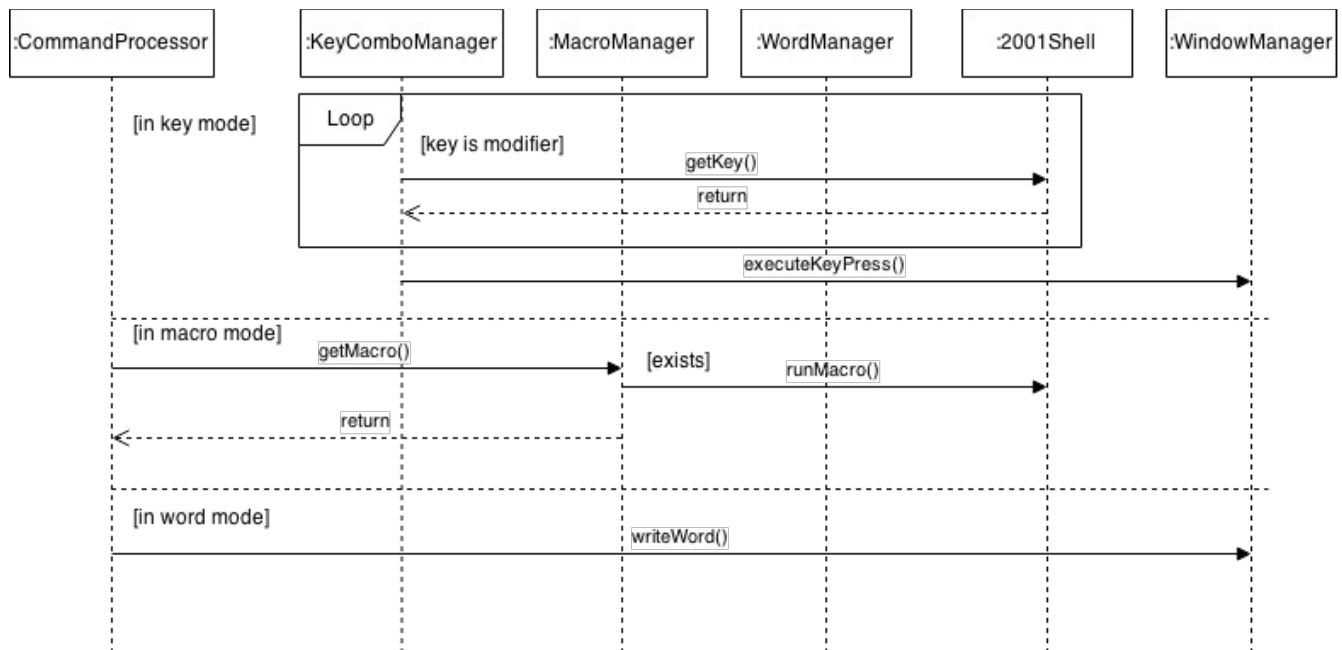
Secuencia del ciclo principal



Secuencia de grabación de macros



Secuencia para correr comandos



Conclusiones y Recomendaciones

- Nos dimos cuenta que varios de los diagramas son versiones mas especificas de otras mas generales. Por ejemplo los de secuencia dependen de los diagramas de actividad y de clases, el de clases, una version refinada del conceptual.
- UML lo obliga a uno a tener un buen diseno del programa, a eliminar ambigüedades pero siempre es mas bonito programar.
- Las herramientas para diagramar UML son toscas, con características innecesarias y molestas. Tienden a tener comportamientos inesperados que lo que hacen es complicar el trabajo.
- Es recomendable hacer el diseno con buen tiempo, las cosas a ultima hora no quedan bien.

Referencias

Miles, R & Hamilton, K. (2006). Learning UML 20 . (1era ed.). Estados Unidos: O'Reilly Media.

Observer (patrón de diseño). (2015). Wikipedia. Retrieved 23 May, 2015, from http://es.wikipedia.org/wiki/Observer_%28patr%C3%B3n_de_dise%C3%B1o%29