



MOUSE PARA MINUSVÁLIDOS

Planteamiento del Problema

El presente trabajo especial de grado, está inspirado en ayudar a aquellas personas que por alguna razón presentan problemas motores o de discapacidad. La idea fundamental es que puedan interactuar con las computadoras mediante un mouse, para que tengan la oportunidad de realizar las cosas que a diario realizan todos, y puedan integrarse más en los procesos de nuestra estructura social. Esto a través de una interfaz que permita a los usuarios hacer uso de las funciones de un mouse, apoyándose en el uso de una webcam y de un micrófono.

Actualmente, existe una diferencia muy notable entre las oportunidades que tienen las personas con problemas motores para interactuar con el mundo computacional, comparadas con las personas que no presentan problemas de salud. Ellos, estarían muy agradecidos de que la tecnología computacional les brindara nuevos avances que les permitan integrarse de manera más activa en los procesos que se llevan a cabo en nuestra estructura social. ¿Será posible desarrollar un tipo de mouse o ratón que les permita a las personas con discapacidades motoras interactuar con la tecnología en computación?

- Fazzino Michelangelo
- Sanchez Oswell

Objetivo General

Realizar un mouse que permita a personas con limitaciones físicas (minusválidas) interactuar con la computadora.

Justificación

Proveer un recurso mediante el cual las personas minusválidas puedan interactuar con las computadoras, parece una gran oportunidad para que puedan hacer uso de las máquinas, tal como lo hacen muchas personas a diario. Se les abrirían así otras posibilidades de participar más activamente, como por ejemplo, en empleos, intercambio de conocimientos entre otros.

Limitaciones y Alcances

El producto va dirigido especialmente hacia personas que tengan movilidad de la cabeza, que puedan hablar y posean el sentido visual. El mouse cumplirá con todas las funciones de un mouse común de dos (2) botones.

El mouse'

El mouse o ratón, fue inventado por Douglas Engelbart en el Centro de Investigación de Stanford en 1963 y empezado a utilizar con fines comerciales por Xerox en los años 70. Esta invención representó un gran avance en el campo de las aplicaciones software ya que facilitó enormemente la comunicación con el usuario, permitiendo el desarrollo de programas basados en entornos gráficos.

Existen muchos tipos de mouse (mecánicos, optomecánicos, ópticos, inalámbricos) que se conectan con el PC de variadas formas (puerto serie, puerto PS/2, etc.). En 1983 Microsoft definió una interfaz de funciones para su mouse (MS-MOUSE) con el MS-DOS que posteriormente se ha establecido como un estándar para todos los controladores de mouse bajo dicho sistema operativo.

La posición que ocupa el mouse en la pantalla se define siempre en coordenadas gráficas, aunque se esté trabajando en modo texto, considerando tantos puntos en la pantalla como permita la tarjeta gráfica. Es importante destacar, que la zona sobre la que trabaja el mouse no coincide con la pantalla real. El mouse se mueve sobre una *pantalla virtual*, que depende del modo de video en el que se esté trabajando, es decir, de la resolución.

Discapacidad²

El término "discapacidad" fue aceptado por la Real Academia Española hace 10 años. Existen otros términos quizás más comunes como "incapacidad", "minusválido" o "inválido" pero estos pueden dar a entender que las personas con discapacidad son personas "sin habilidad", "de menor valor" o "sin valor". En comparación con estas acepciones, la discapacidad tiene que ver con "la disminución de una capacidad en algún área específica", por lo que el uso de este término reconoce que todos los individuos con discapacidades tienen mucho que contribuir a nuestra sociedad. Sin embargo, se hace necesario diferenciar los conceptos de:

Deficiencia: "es toda pérdida o anomalía de una estructura o función psicológica, fisiológica o anatómica". O bien, "es una alteración anatómica o funcional que afecta a un individuo en la actividad de sus órganos."

Discapacidad: "es la consecuencia de una deficiencia, sobre las actividades físicas, intelectuales, afectivo-emocionales y sociales" o también se la puede definir como "toda restricción o ausencia (debido a una deficiencia) de la capacidad de realizar una actividad en la forma o dentro del margen que se considera normal para un ser humano."

La Organización Mundial de la Salud (O.M.S.) define la discapacidad como "la pérdida de la capacidad funcional secundaria, con déficit en un órgano o función, y que trae como consecuencia una minusvalía en el funcionamiento intelectual y en la capacidad para afrontar las demandas cotidianas del entorno social."

Existen cuatro principales tipos de discapacidad, clasificadas según los ámbitos del ser humano al que afectan: Intelectual, Motora, Auditiva y Visual.¹

[en línea]. Disponible en: <<http://atc.urg.es/~jbernier/asignaturas/perifericos/practicas/practica2.pdf>> [25-07-2003].
Discapacidad. Que es?. 2002. Disponible en: <http://www.cilsa.com.ar/web_espanol/discapacidad/que_es.asp>.

Discapacidad motora³

Se define como discapacidad física o motora a la dificultad para realizar actividades motoras convencionales. Esta dificultad puede deberse a múltiples causas, sean congénitas o adquiridas.

Java Media Framework (JMF)⁴

El API (Application Programming Interface) de JMF (Java Media Framework), es una definición de la interfaz utilizada por Java para la utilización de datos multimedia. Como definición de interfaz, JMF indica cómo los proveedores de dichas clases deben encapsularlas y registrarlas en el sistema. Permite a las aplicaciones Java incluir componentes de audio, video, así como capturar, reproducir y decodificar múltiples formatos.

JMF ha sido desarrollada por Sun Microsystems e IBM y no está incluida en las especificaciones de Java 2 o en la máquina virtual, por lo que es necesario obtener el paquete adicional que contiene el JMF para la plataforma que se esté utilizando. JMF, básicamente, realiza tres pasos para el tratamiento de datos multimedia: la adquisición de datos, captura desde un dispositivo físico, lectura de un archivo o recepción desde la red; procesado, aplicación de efectos como filtrado o realces, compresión y/o descompresión, conversión entre formatos; y la salida de datos, presentación, almacenamiento en archivo o transmisión a través de la red.

Java Speech API (JSAPI)^{5,6,7}

El JSAPI, desarrollado por Sun Microsystems en cooperación con compañías de tecnología del discurso (Apple Computer Inc., AT&T, Dragon Systems Inc., IBM Corporation, Novell Inc., Philips Speech Processing, Texas Instruments Incorporated, entre otras), define una interfaz del software que les permite a los

diseñadores aprovechar la tecnología del discurso. El JSAPI define un estándar fácil de usar y soporta dos tecnologías del discurso: el reconocimiento del discurso y el sintetizador del discurso.

El reconocimiento del discurso proporciona la habilidad a las computadoras para escuchar el idioma hablado y determinar lo que se ha dicho. En otros términos, procesa la entrada de audio que contiene el discurso convirtiéndolo en texto. La síntesis del discurso proporciona el proceso inverso de producir el discurso, sintetiza el texto generado por una aplicación, un applet o un usuario y lo convierte en sonido.

La manera principal en que una aplicación controla la actividad de un reconocedor de voz es por medio de las gramáticas. Una gramática es un objeto en el JSAPI que indica qué palabras se espera que diga un usuario y en qué modelos esas palabras pueden ocurrir.

Uno de los mayores beneficios de incorporar la voz en una aplicación es que la voz es natural: las personas encuentran que hablar es fácil, la conversación es una habilidad que la mayoría la domina a temprana edad en la vida.

Proceso Unificado⁸

Para el desarrollo de esta tesis, se utilizó la metodología del Proceso Unificado, que consiste en un proceso de desarrollo de software. Un proceso que busca mediante una serie de actividades transformar los requisitos de un usuario en un sistema software.

El Proceso Unificado, se basa en componentes software interconectados por interfaces bien definidas, además utiliza UML (Lenguaje Unificado de Modelado), para preparar todos los esquemas que se necesitan para describir un sistema. El Proceso Unificado puede definirse en tres frases claves: dirigido por casos de usos, centrado en la arquitectura e iterativo-incremental. Para poder lograr los objetivos se necesitan pasar por cuatro (4) fases de trabajo

Discapacidad. Discapacidad Motora. 2002. Disponible en: <http://www.cilsa.com.ar/web_espagnol/discapacidad/motora.asp>.

⁴ PRADES, Carlos. (2001). Tratamiento multimedia en Java con JMF. [en línea]. Disponible en: <<http://cprades.eresmas.com/Tecnica/programarJMF.pdf>>.

⁵ Sun microsystems. (1998). Java Speech API Programmer's Guide [en línea]. Disponible en: <<http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide.pdf>>.

⁶ Sun microsystems. (1998). Java Speech Grammar Format Specification [en línea]. Disponible en: <<http://java.sun.com/products/java-media/speech/forDevelopers/JSGF.pdf>>.

⁷ Sun microsystems. (1997). Java Speech Markup Language Specification [en línea]. Disponible en: <<http://java.sun.com/products/java-media/speech/forDevelopers/JSML.pdf>>.

⁸ JACOBSON, I., BOOCH, G. y RUMBAUGH, J. El Proceso Unificado de Desarrollo de Software. España, Addison Wesley. 2000. pp. 4-123.

(Inicio, Elaboración, construcción y Transición), además en cada una de ellas se debe cumplir cinco (5) flujos de trabajo (requisito, Análisis, Diseño, Implementación y Prueba).

Desarrollo del prototipo

El desarrollo será descrito en función de los flujos de trabajo y del resultado arrojado por todas las iteraciones que se llevaron a cabo.

Captura de Requisitos

En las primeras iteraciones se consultaron a cinco (5) doctores los cuales argumentaron que "la dificultad motora de los pacientes varía según la discapacidad que cada uno posea". Luego de recopilar la información, se concluyó, según los datos de los médicos, que el 80% de los pacientes conservan la función motora del cuello, el sentido de visión y el lenguaje.

En las siguientes iteraciones, se averiguó sobre las funciones de un mouse común de dos (2) botones (Tabla 4.1). Los requisitos mostrados en la tabla anterior se realizan combinando o no el movimiento del mouse con el hecho de dejar o no presionado cualquiera de los dos (2) botones.

Número	Requisito
	Mover puntero
2	Presionar botón izquierdo
3	Presionar botón derecho
4	Presionar aveces botón derecho
5	Seleccionar objetos
6	Arrastrar objetos con botón izquierdo
7	Arrastrar objetos con botón derecho

Tabla 4.1. Requisitos obtenidos del mouse. Creada por los autores.

Luego se buscó información de cómo los dispositivos hardware (webcam y micrófono) se comunican con el computador y como envían los datos a este. En esta búsqueda, se investigaron varios lenguajes de programación, entre ellos estaban: Visual Basic, Visual C++, C++ y Java. Se encontró que la comunicación estaba contemplada muy bien por las APIs de Java, haciéndola fácil para el programador y permitiéndole tener el control sobre los mismos, lo que originó que Java, fuera seleccionado como lenguaje de desarrollo para esta tesis. Además influyó, la portabilidad que ofrece Java para que el sistema pueda instalarse en diversos sistemas operativos.

En las siguientes iteraciones se buscaron reconocedores de voz para el programa. De estos se encontraron los de idioma inglés, español y chino, de los cuales se seleccionaría el mejor en futuras iteraciones.

Análisis

En las primeras iteraciones del análisis, se determinaron las barreras, inconvenientes y obstáculos para buscar cómo eliminarlos o minimizarlos. Este análisis arrojó los siguientes resultados (Tabla 4.2):

Obstáculo	Inconveniente	Acción Tomada
Realizar un prototipo de casco que permita mover el puntero del mouse	Incomodo para usar, y obliga a la persona a estar unido al computador por medio de cables.	Realizar un prototipo que ofreciera al usuario una forma más libre de interactuar con
Realizar un dispositivo que permita ejecutar los comandos del mouse, mediante la	Engorroso para usar, debido a su funcionamiento.	Ejecutar los comandos del mouse, mediante la voz con la ayuda de un micrófono.
Seleccionar el color que se utilizaría en el análisis de las imágenes.	Problemas en el análisis, debido a que las componentes RGB de los píxeles son afectadas por los colores del espacio físico donde se encuentra el usuario, generando imprecisiones en los cálculos.	Buscar un color tal que minimice el rango de error en los cálculos arrojados por el análisis de las imágenes.

Tabla 42. Obstáculos encontrados. Creada por los autores

En las siguientes iteraciones se procede a desarrollar los casos de uso a partir de los requerimientos obtenidos en las iteraciones iniciales, y para cada uno se describió en detalle: la pre-condición, el flujo principal, los sub-flujos y los flujos alternos. Luego, se elaboró un modelo de casos de uso, que englobaba la funcionalidad total del sistema (Fig. 4.3).

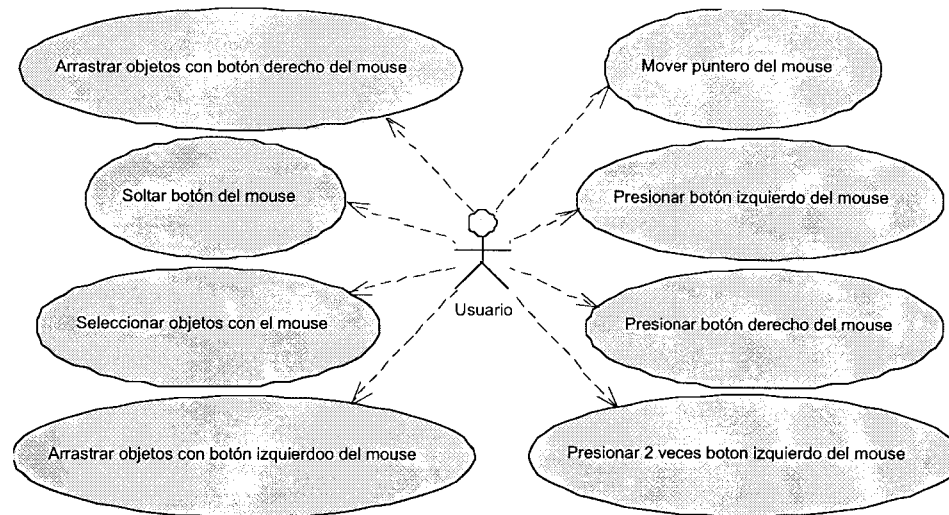


Fig. 4.3 Modelo general de casos de usos. Creado por los autores.

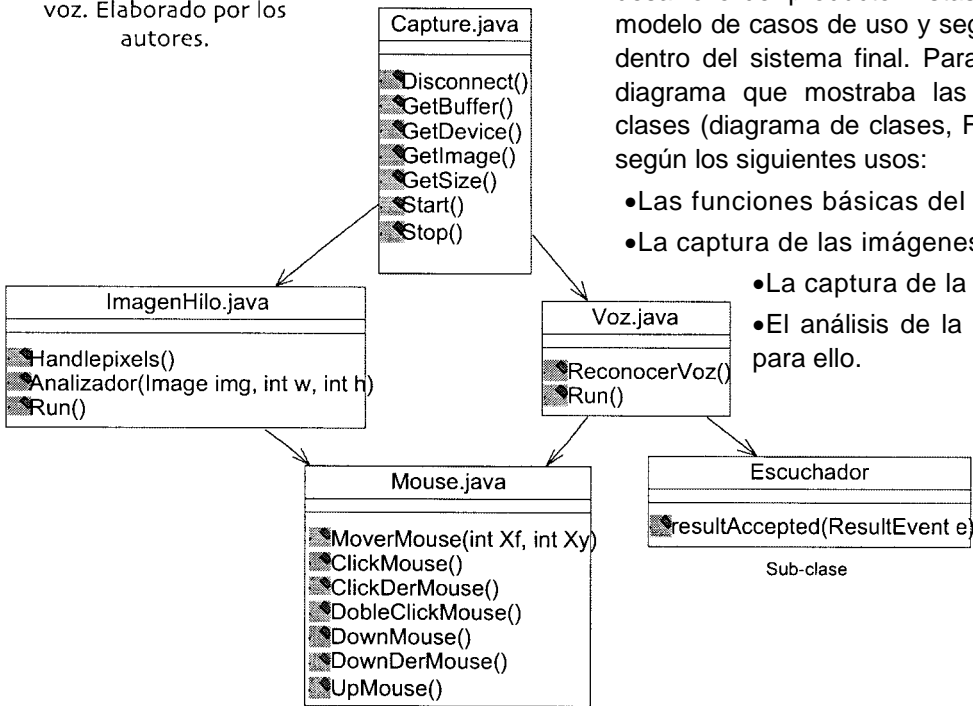
Cada caso de uso de la figura anterior, representa una función que puede ejecutar el usuario dentro del sistema. A continuación se describe brevemente que hace cada uno de ellos (Tabla 4.4).

Caso de uso	Descripción
Mover puntero del mouse.	Este caso de uso mueve el puntero del mouse a la posición
Presionar botón izquierdo del mouse.	Este caso de uso simula el pulsar y soltar el botón izquierdo del mouse una vez (clic izquierdo).
Presionar botón derecho del mouse.	Este caso de uso simula el pulsar y soltar el botón derecho del mouse una vez (clic derecho).
Presionar 2 veces botón izquierdo del mouse.	Este caso de uso simula el pulsar y soltar el botón izquierdo del mouse dos veces seguidas (doble clic).
Seleccionar objetos con el mouse	Este caso de uso simula el seleccionar uno o varios objetos de windows con el mouse.
Arrastrar objetos con botón izquierdo del mouse.	Este caso de uso simula el arrastrar objetos de windows con el botón izquierdo del mouse.
Arrastrar objetos con botón derecho del mouse	Este caso de uso simula el arrastrar objetos de windows con el botón derecho del mouse.
Soltar botón del mouse.	Este caso de uso simula el soltar el botón que este presionado del mouse.

Tabla 4.4 Casos de uso. Creada por los autores.

Diseño

Fig. 4.6. Diagrama de secuencia de comandos de voz. Elaborado por los autores.



Se dio inicio a la identificación de las clases necesarias para el desarrollo del producto. Estas clases fueron clasificadas a partir del modelo de casos de uso y según la funcionalidad que podían ejercer dentro del sistema final. Para cumplir con esta tarea se realizó un diagrama que mostraba las relaciones e interacciones entre las clases (diagrama de clases, Fig. 4.5). Las clases fueron clasificadas según los siguientes usos:

- Las funciones básicas del mouse.
- La captura de las imágenes desde la webcam.
- La captura de la voz, desde el micrófono.
- El análisis de la imagen, por medio de algún algoritmo para ello.

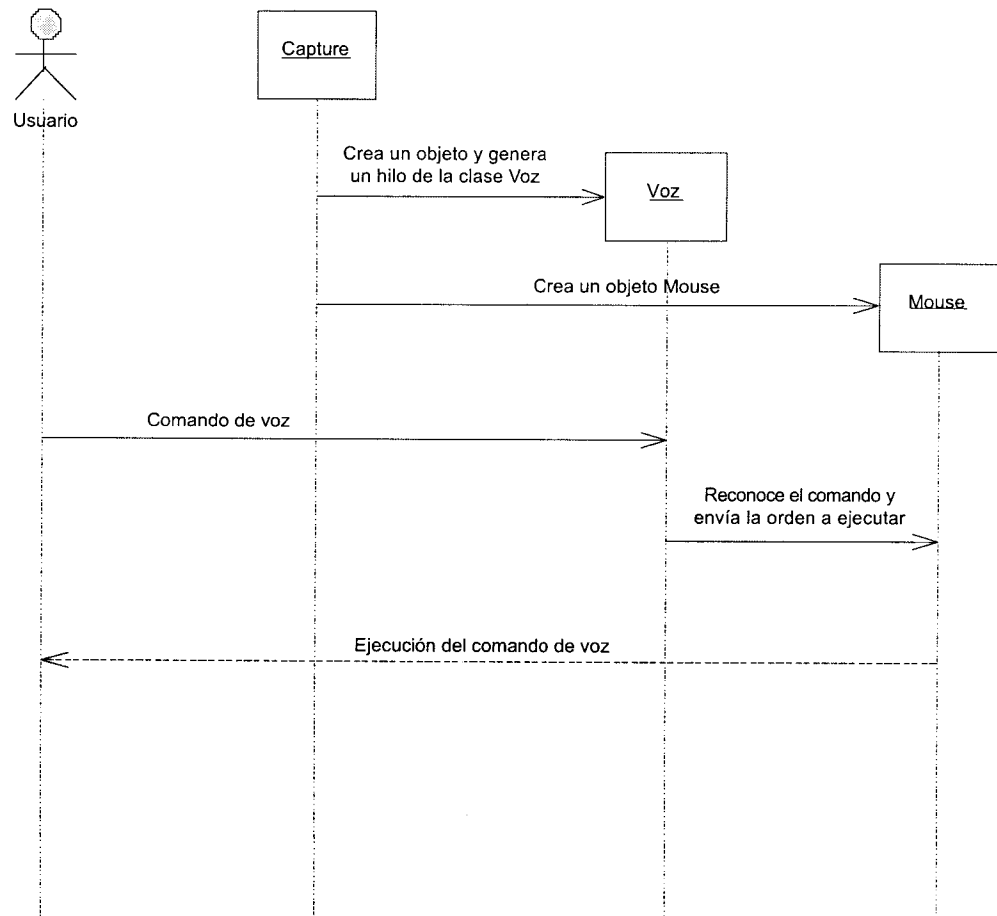


Fig. 4.5. Diagrama de clases del sistema. Cada elemento dentro de una clase es un método de la misma. Elaborado por los autores.

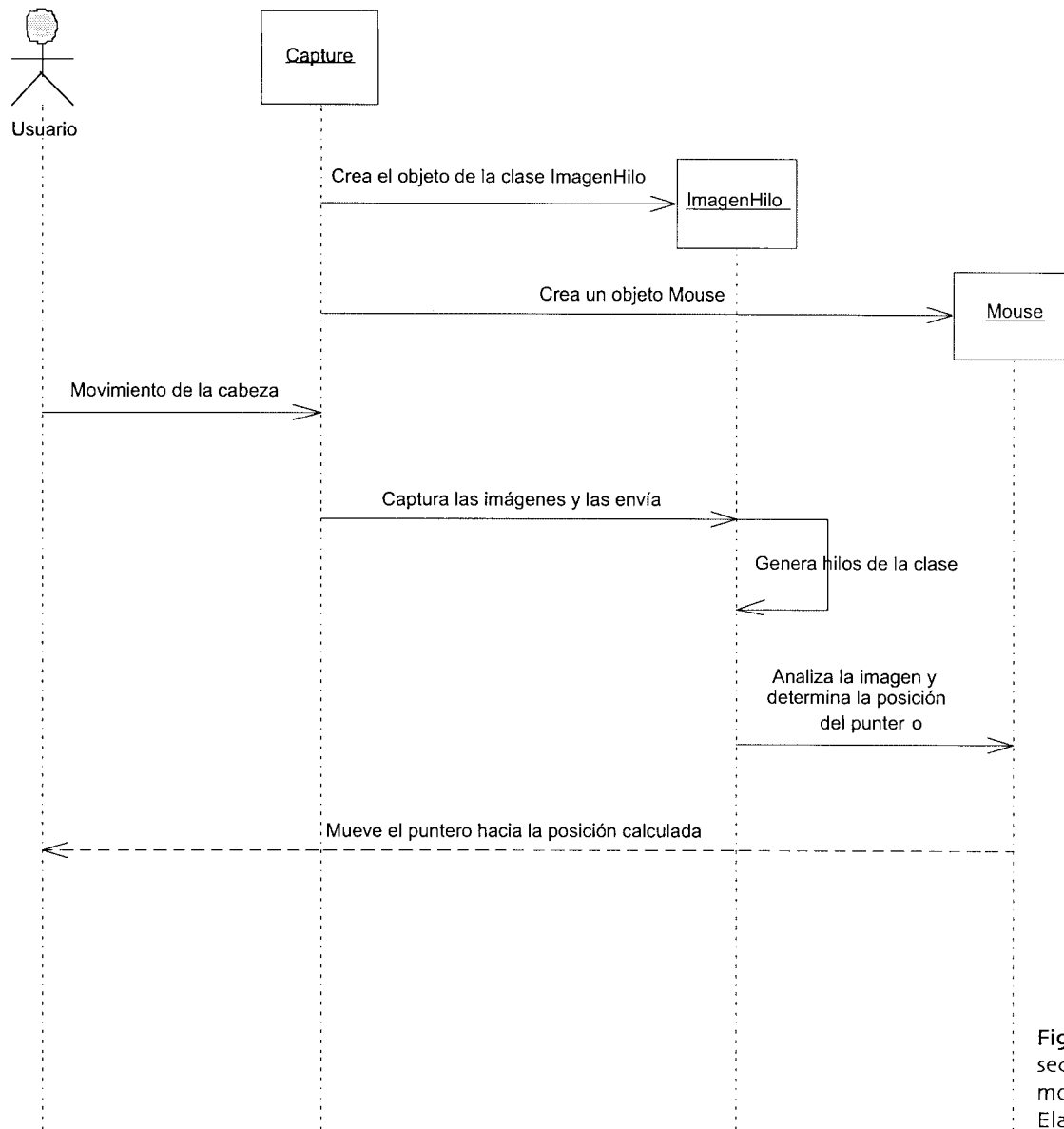


Fig. 4.7. Diagrama de secuencia del movimiento del puntero. Elaborado por los autores.

Además, se realizaron los diagramas de secuencia, para establecer el flujo de información entre los actores y las clases (Fig. 4.6 y Fig. 4.7). Dicho flujo, permitió definir los parámetros que necesitaban enviarse cada una de las clases, obteniéndose una mejor visión de éstas a nivel de funcionamiento.

La figura 4.6, muestra como es la secuencia que se lleva a cabo para ejecutar un comando de voz. Primero la clase *Capture*, crea un objeto de las clases *Voz* y *Mouse*. A partir de aquí, el usuario, puede ejecutar un comando de voz, el cual será recibido y analizado por la clase *Voz*. Inmediatamente, el

resultado es informado a la clase *Mouse*, para que realice la acción y la haga visible al usuario.

La figura 4.7, muestra como es la secuencia que se lleva a cabo para mover el puntero del mouse. Primero la clase *Capture*, crea un objeto de las clases *ImagenHilo* y *Mouse*. A partir de aquí, el usuario puede realizar movimientos de la cabeza, los cuales son capturados y analizados por la clase *ImagenHilo*, que genera varios hilos para realizar esta tarea. El resultado es informado a la clase *Mouse*, para que realice el movimiento y lo haga visible al usuario.

Implementación

Una vez terminado el modelo de casos de uso, el diagrama de clases y el diagrama de secuencias, se procede a la culminación de la arquitectura. A partir de aquí, se definieron las clases que serían implementadas físicamente. Ellas fueron: *Capture.java*, *Voz.java*, *Mouse.java* e *ImagenHilo.java*, según la funcionalidad que debían aportar al sistema final.

Captura de imágenes: Esta tarea está delegada a la clase *Capture.java*, que por medio de sus métodos puede obtener imágenes desde una webcam. El primer paso para la captura de las imágenes es buscar los dispositivos hardware que están instalados en el equipo de los cuales se pueden recibir imágenes, entre ellos: escáneres, webcam, cámaras digitales, etc. Luego de obtener la lista de dispositivos, se selecciona aquel referido a una webcam, en caso de que no exista ninguno el sistema emite un mensaje para indicar la situación. Posteriormente, se establecen los parámetros necesarios para la comunicación con la webcam, es decir, la creación de la fuente de datos, el tamaño de las imágenes, el tipo de formato multimedia a procesar, etc. Se crea además un player, que recibe el flujo de información y proporciona la fuente de datos, para poder procesarla y analizarla en el momento indicado.

Una vez establecida la comunicación con la webcam, por medio del player, se procede a convertir el flujo de información en imágenes consecutivas, que son enviadas hacia un analizador para que las examine (*ImagenHilo.java*). Cada vez que el sistema por medio de la webcam obtiene una nueva imagen, ésta se envía de inmediato al analizador. Durante todo el tiempo que el sistema está activo, se repite el ciclo de capturar y analizar las imágenes.

Procesamiento de imágenes: Esta labor es realizada por la clase *ImagenHilo.java*. El procedimiento se inicia cuando la clase *Capture.java* envía una imagen para ser analizada. Dicha imagen es segmentada en nueve (9) recuadros, que indican una dirección de movimiento (Fig. 4.9). Luego por cada dirección, se genera un hilo para que analice su porción de la imagen. El análisis que realiza cada hilo consiste en buscar dentro de la porción de la imagen que le corresponde, los píxeles de color azul que se encuentran allí y llevar un conteo de los mismos. Cuando todos los hilos terminen su conteo, se verifica cuál de ellos encontró la mayor cantidad de píxeles. Finalmente, el puntero del mouse se mueve en velocidad constante hacia esa dirección, con la ayuda de la clase *Mouse.java*.

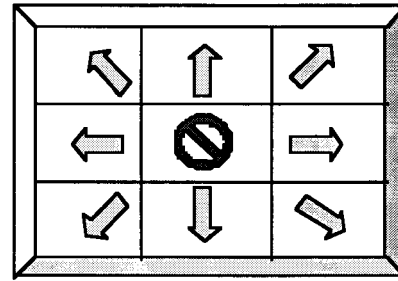


Fig. 4.9. Direcciones de movimientos. Elaborado por los autores

Procesamiento de la voz: Esta labor es realizada por la clase *Voz.java*, y se define dentro de esta clase, una gramática que contiene la sintaxis de las palabras que serán permitidas captar del usuario. Dicha gramática necesita que un reconocedor de voz esté activo para que reciba el sonido desde un micrófono. Este reconocedor de voz no forma parte de este sistema, por lo cual el equipo debe contar con uno para que el reconocimiento de palabras pueda funcionar. Se encontraron varios reconocedores en idiomas diferentes. Luego de realizar las pruebas pertinentes, se determinó que el que mejor funcionaba era el de idioma inglés, el cual es el utilizado por este sistema.

El sistema cuando es iniciado genera una lista con todos los reconocedores de voz que tenga el equipo y luego selecciona uno que esté en el idioma inglés. En caso de que la lista esté vacía o no exista un reconocedor en inglés, el sistema envía un mensaje e indica la situación. Una vez completada la selección del reconocedor, se coloca en estado *escuchando* y se le asignan los recursos requeridos para que funcione; inmediatamente la gramática es activada y colocada en estado activo para ser usada por el reconocedor.

El sistema genera un hilo del programa principal que se encargará de procesar todos los comandos de voz que el usuario proporcione. Este hilo permanece activo durante el tiempo que el sistema se mantiene en ejecución. Por lo tanto, el usuario puede introducir el comando de voz que desee ejecutar en cualquier momento, una vez iniciado y activado el sistema. Estos comandos de voz son procesados por el reconocedor, el cual detecta la expresión y genera un evento por medio de la clase *Mouse.java* para que el sistema operativo lo procese (estado *procesando*). Si el reconocedor encuentra que la palabra no satisface la gramática activa, se coloca nuevamente en estado *escuchando*, en espera de un nuevo comando de voz.

Manejo del mouse: Esta labor es desempeñada por la clase *Mouse java*, que permite ejecutar todas las operaciones desempeñadas por un mouse estándar de dos (2) botones. Para la realización de éstas, es necesario el apoyo de las clases "java.awt.GraphicsEnvironment", "java.awt.event.InputEvent", y "java.awt.Robot".

La primera se utiliza para capturar la configuración de la pantalla principal o de fondo, dicha configuración obtiene mucha información sobre la misma, como por ejemplo los objetos que se encuentran dentro de ella o la cantidad de color en píxeles que soporta; para este sistema es vital debido a que informa sobre la resolución o las dimensiones dentro de las cuales será ubicado el puntero. En ocasiones será necesario saber si la pantalla gráfica principal contiene objetos, como por ejemplo, ventanas gráficas, íconos, botones, entre otros, y así poder interactuar con ellos.

La segunda clase en conjunto, con la tercera citada anteriormente, generan eventos que permiten ser atrapados por el sistema operativo para su posterior ejecución. Estos eventos pueden ser orientados hacia el teclado o el mouse. Si están orientados al mouse, se realizan acciones como: dejar presionado un botón por un período de tiempo, liberar un botón que haya sido presionado, mover el puntero hacia una posición de la pantalla, entre otras. El sistema operativo al escuchar el evento generado por estas clases, lo recibe y lo procesa sobre los objetos que se encuentren activos dentro de la pantalla principal. Para este sistema, sólo se hace uso de los eventos orientados al mouse.

Pruebas

Para la realización de este trabajo, se realizaron pruebas de caja negra mediante los casos de prueba, con la finalidad de verificar cada uno de los casos de uso del sistema. Todas estas arrojaron los resultados señalados en los propios casos de uso.

Condiciones de uso

A continuación se exponen las condiciones ideales para que el producto funcione correctamente, estas fueron obtenidas durante el periodo de desarrollo del proyecto, las cuales son:

- El equipo donde va a funcionar el sistema debe estar en buenas condiciones a nivel de hardware y software.

- La webcam debe estar enfocada hacia la cara de la persona que va a utilizar el mouse de manera que ésta quede frente al lente de la cámara.
- El espacio físico donde se encuentre el usuario debe estar iluminado con luz de neón blanca.
- El espacio físico donde se encuentre el usuario debe prescindir de colores iguales o cercanos al color azul.
- Es importante destacar que en el lugar físico donde se encuentre el usuario, debe mantenerse un mínimo de ruido para que los comandos de voz puedan ser procesados correctamente.

Conclusión

Con este producto, se aumentan las posibilidades de que las personas minusválidas puedan encontrar trabajos, enseñar a otras personas, realizar cosas cotidianas como revisar el correo, realizar un documento digital, conectarse a Internet, entre otras. Al mismo tiempo se les brinda la oportunidad de integrarse a la tecnología y a que no se sientan rechazados por la misma.

La facilidad de uso y el confort están contemplados en este sistema, debido a que las acciones para manejarlo son sencillas y de fácil aprendizaje. Esto le da un valor agregado al sistema permitiendo que su uso se extienda a otras personas y aplicaciones.

Este trabajo puede ser el inicio para que otras personas se motiven a realizar otros productos que ayuden a las personas con problemas motores a que poco a poco formen parte del apasionante mundo informático.