



COMPUTADORES III

VIRTUAL CYBERTECH: USER REQUIREMENTS



UNIVERSIDAD POLITÉCNICA DE MADRID
**ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS INDUSTRIALES**

EQUIPO YELLOW:

Marta Dorado (Project Manager), Álvaro López, Antonio Díez, Carlos Sampedro, Francisco Suárez, Marie Destarac y Ricardo Espinoza.

Índice

0. Seguimiento del documento	3
1. Introducción.....	4
1.1 Propósito.....	4
1.2 Ámbito del sistema.....	4
1.3 Definiciones, acrónimos y abreviaturas	5
1.4 Referencias	5
1.5 Visión general del documento	6
2. Descripción general.....	7
2.1 Perspectiva del producto.....	7
2.2 Funciones del producto	7
2.3 Características de los usuarios	7
2.4 Suposiciones y dependencias	8
2.5 Requisitos futuros	8
3. Requisitos Específicos	9
3.1 Requisitos de usuario	11
3.2 Requisitos físicos	23
3.2.1 Simular Robot Kephra III	23
3.2.2 Condiciones de la Tierra.....	24
3.2.3 Condiciones del laberinto	25
3.3 Requisitos de la simulación.....	28
4. Diagramas del producto	32
4.1 Casos de uso	32
4.2. Interfaz de usuario	48

0. Seguimiento del documento

Histórico de modificaciones

Versión Número	Fecha <<mm/dd/aa>>	Cambios	Autor	Revisor	Secciones afectadas
1	04/13/12	Creación del documento	Equipo Yellow	Guadalupe Sánchez	Todas
2	05/07/12	Se agregaron los casos de uso y el bosquejo de la GUI	Equipo Yellow	Guadalupe Sánchez	Diagramas del producto
3	06/23/12	Modificación y actualización de casos de usos. Actualización de la descripción de la interfaz de usuario.	Equipo Yellow	Guadalupe Sánchez	Diagramas del producto e interfaz de usuario.

1. Introducción

1.1 Propósito

La finalidad de este documento es especificar los requisitos que debe cumplir el software a desarrollar, siguiendo las bases de Cybertech 2012 y las necesidades del cliente. Este documento servirá de enlace entre lo que el cliente quiere y los desarrolladores realicen, para su posterior validación.

1.2 Ámbito del sistema

El sistema a diseñar se ha denominado *Virtual CyberTech*, en adelante VCT, y su función es simular un entorno para la competición *Cybertech*, que se realiza anualmente entre grupos de distintas universidades y es organizada por la División de Ingeniería de Sistemas y Automática y la asociación de estudiantes Reset de la Escuela Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid.

Esta competición se basa en el diseño y construcción de Robots móviles que deben superar las siguientes pruebas:

- Rastreador
- Laberinto
- Final (Rastreador y Laberinto)

El sistema VCT busca desarrollar un entorno virtual donde los participantes del concurso puedan llevar a cabo simulaciones en tiempo real de los códigos desarrollados para su robot en un entorno que tiene las mismas características y limitaciones reales del concurso.

Los objetivos principales del sistema se presentan a continuación:

- El sistema deberá proveer los recursos necesarios para realizar la competencia CyberTech 2012 en un entorno virtual.
- El entorno de simulación se basará en el Virtual Robot Experimental Platform (V-REP)

- El robot a usar es el Kephera III.
- El ambiente de integración se realizará con la herramienta de IBM, Rational Software Architect.
- La interfaz se realizara a través del software para desarrollar GUI de Nokia QT SDK

Lo que se pretende es que los participantes sean capaces de probar los algoritmos de control que implementarán en sus robots incluso antes de la construcción de los mismos, lo que sugiere una gran ventaja, ya que la experiencia en los eventos previos indica que la mayor parte de tiempo que requiere el concurso se invierte en la construcción del hardware.

El alcance de esta primera versión se centra en el desarrollo del entorno de simulación para laberintos, simulando un robot Kephera III, y en donde es posible construir múltiples laberintos según las condiciones de la competición.

1.3 Definiciones, acrónimos y abreviaturas

Con la finalidad de mantener claro en todo momento los términos que se usan en estas especificaciones, se indican a continuación todas las definiciones, acrónimos y abreviaturas utilizadas:

- Virtual CyberTech: VCT
- Virtual Robot Experimental Plataform: V-REP
- Rational Software Architect: RSA.
- Kephera III: KIII
- Universidad Politécnica de Madrid: UPM

1.4 Referencias

Se tomaron como base los siguientes documentos:

- Especificación de requisitos según el estándar IEEE 830-1998

- Requisitos iniciales VCT_V0.1.docx con fecha 26/03/12.
- Component Approach for Real-time and Embedded. Documento D2.2-1, versión 3.0, con fecha 21/02/07.

1.5 Visión general del documento

En el apartado 2 se presenta la descripción general del producto a realizar, así como las funciones que éste tendrá. En el apartado 3 se muestra listados los requisitos y estos se han organizado en tres grupos: requisitos de usuario, requisitos físicos y requisitos de la simulación. Cada requisito se presenta en una tabla donde se realiza la descripción, limitaciones y otras características, dentro de las que destaca el tipo de requisito que es, que puede ser funcional, no funcional y de dominio. Para una mejor comprensión de la organización de los requisitos, se ha incluido un índice al inicio del apartado 3.

En el apartado 4 se muestran los casos de uso del producto y se presenta la interfaz de usuario, explicando de forma general cada parte que la compone.

2. Descripción general

2.1 Perspectiva del producto

El VCT es un producto independiente en el sentido de que no pertenece a un sistema mayor, pero su funcionamiento está condicionado al sistema operativo de Windows. El producto a desarrollar será de tipo ejecutable, por lo que el usuario no deberá tener conocimientos en informática para ponerlo en funcionamiento.

2.2 Funciones del producto

El VCT provee un entorno de simulación para la competición de CyberTech y permitirá la creación o selección de un entorno, la ejecución del código del KIII que el usuario ha creado anteriormente, la visualización del comportamiento del KIII en el entorno, la posibilidad de pausar la simulación y guardarla para su posterior visualización. El VCT facilita la comprobación del desempeño del KIII en el laberinto seleccionado sin necesidad de contar con el entorno real, lo que reducirá el tiempo que el usuario emplee en mejorar el algoritmo de control.

2.3 Características de los usuarios

Siguiendo las bases del concurso CyberTech, al menos un usuario por grupo del producto VCT, será alumno de la UPM, por lo que se entiende que cuenta con conocimientos mínimos de informática. Se sobreentiende que habrá al menos un miembro del grupo que cuenta con estudios de bachillerato y experiencia técnica suficiente para construir y programar un robot.

2.4 Suposiciones y dependencias

Sistema operativo: el VCT será diseñado para trabajar sobre el sistema operativo Windows.

Robot a usar: el VCT simula el comportamiento del robot KIII en los diferentes laberintos disponibles, tomando como base sus características físicas y su lenguaje de programación.

Entorno de simulación: el VCT se desarrollará en V-REP.

Número de robots a simular: se supone que el VCT será capaz de soportar la simulación de un robot moviéndose sobre el entorno seleccionado.

Interfaz de Usuario: se desarrolla con la herramienta QT SDK.

2.5 Requisitos futuros

Las mejoras que pueden implementarse es que el VCT funcione en diferentes plataformas, simule más de un robot dentro del entorno, acepte diferentes tipos de robots al ingresar las características físicas de estos y realice la simulación en tiempo real.

3. Requisitos Específicos

Los requisitos se han organizado en tres grupos: requisitos de usuario, requisitos físicos y requisitos de la simulación. Cada requisito se presenta en una tabla donde se realiza la descripción, limitaciones y otras características. A continuación se muestra el esquema de división de los requisitos:

1. Requisitos de Usuario

- 1.1 Programa ejecutable
- 1.2 Stand-alone
- 1.3 Idioma de la aplicación
- 1.4 Mostrar interfaz de la aplicación
 - 1.4.1 Mostrar cuadro de ingreso de código
 - 1.4.2 Mostrar cuadro de controles con los siguientes acciones:
 - 1.4.2.1 Botón Start
 - 1.4.2.2 Botón Pausa
 - 1.4.2.3 Botón Stop
 - 1.4.2.4 Ejecutar Paso a Paso (plus)
 - 1.4.2.5 Variar Velocidad de Simulación
 - 1.4.2.6 Botón Guardar simulación
 - 1.4.2.7 Abrir Simulación
 - 1.4.2.8 Botón Cerrar Aplicación
 - 1.4.2.9 Slider zoom
 - 1.4.2.10 Seleccionar laberinto
 - 1.4.3 Mostrar cuadro de simulación, que consta de:
 - 1.4.3.1 Ventana de entorno
 - 1.4.3.2 Ventana de gráficos (posición y velocidad)
 - 1.4.3.3 Cuadro de datos (aceleración estado de sensores)
 - 1.4.4 Ventana de status.
- 1.5 Instalador
 - 1.5.1 Incluir Librerías
 - 1.5.2 Copia de archivos necesarios
 - 1.5.3 Crear acceso directo
- 1.6 Almacenar
 - 1.6.1 Datos
 - 1.6.2 Laberintos
 - 1.6.3 Programa Robot

- 1.6.4 Simulaciones
- 1.7 Tipos de usuario
 - 1.7.1 Usuario básico
 - 1.7.2 Administrador
- 2. Requisitos Físicos
 - 2.1 Simular Robot Kephera III
 - 2.1.1 Velocidad máxima
 - 2.1.2 Peso
 - 2.1.3 Dimensiones
 - 2.1.4 Sensores
 - 2.2 Condiciones de la Tierra
 - 2.2.1 Gravedad
 - 2.2.2 Rozamiento
 - 2.3 Condiciones del Laberinto
 - 2.3.1 Altura mínima de paredes
 - 2.3.2 Distancia entre paredes
 - 2.3.3 Las paredes forman ángulo recto entre ellas y con el suelo.
 - 2.3.4 El suelo deber ser de color blanco.
 - 2.3.5 No deben existir islas.
 - 2.3.6 Debe tener una entrada y una salida.
- 3. Requisitos de la simulación
 - 3.1. Velocidad
 - 3.1.1. Simular con velocidad cercana al tiempo real.
 - 3.1.2. Variar velocidad de simulación.
 - 3.2. Cámara
 - 3.2.1. Presentar vista superior dentro del cuadro de simulación
 - 3.2.2. Presentar vista en perspectiva isométrica dentro del cuadro de simulación.
 - 3.2.3. Presentar vista en primera persona.
 - 3.2.4. Variar el zoom.
 - 3.3. Ubicación de Objetos
 - 3.3.1. Situar inicialmente al robot en cualquier parte del laberinto.
 - 3.3.2. Ubicar las paredes del laberinto a gusto del usuario (fijas una vez iniciada)
 - 3.3.3. Generar laberintos aleatoriamente previa solicitud del usuario.

3.1 Requisitos de usuario

Nº	1.1	Nombre Requisito:	Programa ejecutable	Fecha:	13/04/2012	Persona responsable:	Ricardo Espinoza
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El simulador deberá ejecutarse desde un solo archivo '.exe' sin necesidad de programas adicionales. Esta aplicación lanzará tanto el entorno de simulación como la interfaz de usuario, la consola de edición, que permite realizar las operaciones de inserción de código en el robot.							
Criterio de verificación: Al realizar doble click sobre el ejecutable, este se abre y lanza la aplicación.							
Limitaciones: Este programa solo funcionará en el sistema operativo Windows Vista/7.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.2	Nombre Requisito:	Stand-Alone	Fecha:	13/04/2012	Persona responsable:	Ricardo Espinoza
Prioridad:			Alta	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: La aplicación dispondrá de una serie de librerías asociadas que se deberán cargar automáticamente al inicio de la ejecución sin la intervención del usuario.							
Criterio de verificación: Al iniciar la aplicación deberán estar disponibles todas las aplicaciones, sin que el usuario deba intervenir directamente.							
Limitaciones: Contará con las librerías necesarias para el correcto funcionamiento de las utilidades de la aplicación. Para ello no debe ser necesaria una conexión a Internet.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.3	Nombre Requisito:	Idioma de la aplicación	Fecha:	13/04/2012	Persona responsable:	Ricardo Espinoza
Prioridad:			Media	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La interfaz de la aplicación debe tener como idioma principal el inglés.							
Criterio de verificación: No existirán palabras en otro idioma							
Limitaciones: Ninguna							
Tipo de requisito: Dominio.							
Estado: Aprobado							

Nº	1.4	Nombre Requisito:	Mostrar interfaz de la aplicación	Fecha:	13/04/2012	Persona responsable:	Ricardo Espinoza
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La aplicación deberá contar con una interfaz gráfica a través de la cual el usuario pueda utilizar todas las herramientas y utilidades disponibles.							
Criterio de verificación: Al ejecutarse la aplicación se debe acceder a la interfaz.							
Limitaciones: Estará dividida en 4 partes (controles, simulación, código y estatus)							
Tipo de requisito: Funcional.							
Estado: Aprobado.							

Nº	1.4.1	Nombre Requisito:	Mostrar cuadro de ingreso de código	Fecha:	13/04/2012	Persona responsable:	Ricardo Espinoza
Prioridad:			Alta	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: La interfaz del usuario contará con un cuadro dedicado a la escritura y edición del código del programa que se ejecutará en el robot durante la simulación.							
Criterio de verificación: Al acceder a la interfaz debe estar disponible el cuadro de código.							
Limitaciones: La longitud del código viene limitada por las capacidades del controlador del Robot Kephera III							
Tipo de requisito: Funcional.							
Estado: Desarrollo							

Nº	1.4.2	Nombre Requisito:	Mostrar cuadro de controles	Fecha:	13/04/2012	Persona responsable:	Antonio Díez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El simulador debe tener un cuadro de botones desde el que sea posible visualizar e interactuar con ellos. Estos botones deberán estar juntos, de tal forma que conformen un conjunto fácilmente identificable.							
Criterio de verificación: Al lanzar la aplicación deben aparecer dichos botones en la interfaz de esta.							
Limitaciones: Los botones deben de tener un tamaño mínimo de tal forma que sea posible mostrarlos por pantalla de tal forma que en ellos esté escrita su etiqueta, que a su vez corresponderá con su funcionalidad.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.1	Nombre Requisito:	Botón Start	Fecha:	13/04/2012	Persona responsable:	Antonio Díez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Dentro del cuadro definido en el punto 1.4.2 se debe encontrar un botón con la siguiente etiqueta: "Start". Dicho botón debe ser capaz de arrancar la simulación, mostrando al usuario el resultado producto de los valores y código empleado. Excepto en caso de que esté activada la opción "Ej. por pasos" que entonces supondrá la ejecución de una línea de código.							
Criterio de verificación: Al presionar sobre el botón la simulación empieza a ejecutarse.							
Limitaciones: Sólo se podrá tener en ejecución una simulación, de tal forma que el botón ignorará sus llamadas mientras haya una simulación en curso.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.2	Nombre Requisito:	Botón Pause	Fecha:	13/04/2012	Persona responsable:	Antonio Díez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Dentro del cuadro definido en el punto 1.4.2 se debe encontrar un botón con la siguiente etiqueta "Pause". Dicho botón debe ser capaz de pausar la simulación, dejando la simulación pausada hasta que se pulse otra vez este botón o en su defecto el botón STOP (ver requisito 1.4.2.3).							
Criterio de verificación: Al pulsarse este botón la simulación debe hacer un alto.							
Limitaciones: Al ser únicamente posible tener en funcionamiento una simulación, sólo debe pausar una simulación							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.3	Nombre Requisito:	Botón Stop	Fecha:	13/04/2012	Persona responsable:	Antonio Díez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Entre los botones definidos por el cuadro de controles se encontrará uno con la etiqueta "Stop", éste deberá ser capaz de detener completamente la simulación, de modo que no sea posible volver a ella ni continuarla de ninguna manera. Excepto en caso de que esté activada la opción "Ej. Por pasos" que entonces supondrá la salida de este modo.							
Criterio de verificación: Al pulsar el botón de la simulación se detiene, de tal forma que es imposible continuar con ella.							
Limitaciones: Al ser únicamente posible tener en funcionamiento una simulación, sólo debe detener una simulación.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.4	Nombre Requisito:	Ejecutar paso a paso	Fecha:	13/04/2012	Persona responsable:	Antonio Díez
Prioridad:			Baja	Estabilidad del Documento	Puede cambiar	Release:	1.0
Descripción: En el interior del cuadro especificado en el punto 1.4.2 se encontrará un botón con la etiqueta “Ej. por pasos”. Al ser pulsado se inicia la ejecución paso a paso, quedando habilitado el botón “Start” y “Stop”. Cada pulsación de “Start” ejecutará una línea de código, el botón “Stop” detendrá la ejecución paso a paso.							
Criterio de verificación: La simulación debe pararse en la primera pulsación y al pulsarse las siguientes veces, debe observarse como el estado de la simulación cambia.							
Limitaciones: Al ser únicamente posible tener en funcionamiento una simulación, sólo se podrá ejecutar una única simulación paso a paso.							
Tipo de requisito: Funcional.							
Estado: Opcional							

Nº	1.4.2.5	Nombre Requisito:	Variar velocidad de simulación	Fecha:	13/04/2012	Persona responsable:	Antonio Díez
Prioridad:			Media	Estabilidad del Documento	Puede cambiar	Release:	1.0
Descripción: Deberá de existir en el cuadro del punto 1.4.2 un slider para variar la velocidad de simulación, o en su defecto dos botones, uno con la etiqueta “+” y otro con la etiqueta “-” que sean capaces de aumentar y reducir la velocidad de ejecución de la simulación, respectivamente.							
Criterio de verificación: Al variar la posición del cuadro del slider o al pulsar los botones se debe aumentar o disminuir la velocidad de simulación, de forma que los tiempos medidos en la realidad para la ejecución de las simulaciones varíe en al menos 10 segundos.							
Limitaciones: Debe existir una velocidad mínima de tal forma que sea imposible pausar la simulación a costa de pulsar las suficientes veces el botón de reducir la velocidad o llevar el slider a la posición extrema.							
Tipo de requisito: Funcional.							
Estado: Opcional							

Nº	1.4.2.6	Nombre Requisito:	Botón guardar simulación	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Dentro del cuadro de controles se encontrará un botón que permita guardar la simulación. Dicho botón creará la primera vez una carpeta con la denominación “Laberinto-Código de robot” permitiendo al usuario el almacenamiento de la última simulación realizada en la ubicación especificada.							
Criterio de verificación: Al presionar sobre el botón de guardar simulación, se crearán los archivos necesarios para la recuperación de la simulación. También se creará la carpeta del punto anterior.							
Limitaciones: No se debe permitir al usuario modificar los archivos creados por cada simulación.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.7	Nombre Requisito:	Botón abrir simulación	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Dentro del cuadro de controles, deberá existir un botón que permita al usuario abrir la simulación. Este botón permitirá elegir entre las simulaciones que se encuentren almacenadas y cargar la deseada para ser visualizada y/o modificada.							
Criterio de verificación: Al realizar click sobre el botón de abrir simulación y elegir la que se desea ver, se cargará dicha simulación, incluyendo laberinto y código.							
Limitaciones: Sólo se permite abrir una simulación a la vez.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.8	Nombre Requisito:	Botón cerrar aplicación	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La interfaz gráfica dispondrá de un botón en la esquina superior derecha en forma de aspa "X" cuya función será la de cerrar la aplicación							
Criterio de verificación: Al presionar el botón la aplicación se cerrará.							
Limitaciones: Ninguna							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.2.9	Nombre Requisito:	Slider Zoom	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Media	Estabilidad del Documento	Puede cambiar	Release:	1.0
Descripción: En el cuadro de controles existirá un slider para ajustar el zoom donde el usuario podrá variar este, aumentándolo o disminuyéndolo según el lado hacia el que realice el desplazamiento, siendo hacia el lado derecho donde se produzca el aumento y hacia el lado izquierdo el efecto contrario.							
Criterio de verificación: Al deslizar el slider, la imagen del cuadro de simulación aumentará o disminuirá de tamaño en la perspectiva que se encuentre seleccionada.							
Limitaciones: El zoom tiene un rango limitado.							
Tipo de requisito: Funcional.							
Estado: Opcional							

Nº	1.4.2.10	Nombre Requisito:	Seleccionar laberinto	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La aplicación dispondrá de un botón que abrirá el explorador de directorios donde se encuentran los diferentes laberintos disponibles en disco (predefinidos, creados por el usuario), para que el usuario cargue uno de ellos, el cual será visualizado en la ventana de simulación.							
Criterio de verificación: Al presionar sobre el botón de selección de laberinto, se mostrará en un explorador el directorio donde estén almacenados los laberintos, y se permitirá la selección de uno.							
Limitaciones: Sólo se podrá seleccionar un laberinto. Sólo se visualizarán los archivos del tipo 'laberinto'.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.3	Nombre Requisito:	Mostrar cuadro de simulación	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La aplicación dispondrá de una región definida en la cual aparecerán 3 elementos: Ventana de entorno (1.4.3.1), Ventana de gráficos (1.4.3.2) y Cuadro de datos (1.4.3.3). Este proceso se ejecutará por solicitud de la aplicación y será el encargado de presentar visualmente la información de la simulación al usuario.							
Criterio de verificación: Al ejecutar la aplicación aparece el cuadro de simulación, además en los casos de simulación detenida o simulación pausada mantendrá en pantalla la última información disponible de la simulación. Durante la simulación deberá actualizarse después de su llamada en cada paso de la simulación.							
Limitaciones: Fijadas por 1.4.3.1, 1.4.3.2 ,1.4.3.3							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.3.1	Nombre Requisito:	Ventana de entorno	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Es la región del simulador encargada de realizar el renderizado 3D. En ella se mostrara de forma visual el estado de la simulación así como los objetos que hacen parte de ella: Robot, paredes (Laberinto), suelo, etc.							
Criterio de verificación: Cada vez que se requiere mostrar el cuadro de simulación (1.4.3) la ventana de entorno muestra la información gráfica.							
Limitaciones: Se definirá un compromiso entre la velocidad de la simulación y el tamaño de la ventana con el fin que la aplicación sea lo más rápida posible y en la ventana de entorno se pueda apreciar cómodamente el renderizado 3D.							
Tipo de requisito: Funcional.							
Estado: Aprobado.							

Nº	1.4.3.2	Nombre Requisito:	Ventana de gráficos	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Es la región del simulador encargada de mostrar una gráfica que se irá alimentando de datos a medida que la simulación transcurre. En esta ventana se mostrará la posición y velocidad del Kephra III en función del tiempo.							
Criterio de verificación: Durante la simulación se muestra y actualiza esta ventana.							
Limitaciones: Solo se mostrará información de posición y velocidad del robot.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.4.3.3	Nombre Requisito:	Cuadro de datos	Fecha:	13/04/2012	Persona responsable:	Marie Andréé Destarac
Prioridad:			Media	Estabilidad del Documento	Puede cambiar	Release:	1.0
Descripción: En el cuadro de simulación se mostrará el valor de tiempo, coordenadas (x,y) del robot, velocidad del robot, y estado de los sensores de éste mientras dura la simulación.							
Criterio de verificación: El tiempo, coordenadas (x,y) del robot, velocidad del robot, estado de los sensores, así como la aceleración del robot aparecen como datos visibles durante cada simulación requerida por el usuario.							
Limitaciones: Los datos mostrados no pueden ser modificados por el usuario durante la simulación.							
Tipo de requisito: Funcional							
Estado: Aprobado							

Nº	1.4.4	Nombre Requisito:	Ventana de estatus de la simulación	Fecha:	13/04/2012	Persona responsable:	Ricardo Espinoza
Prioridad:			Media	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La interfaz del usuario contará con una ventana en donde se podrá seguir la evolución de la simulación							
Criterio de verificación: Al acceder a la interfaz debe estar disponible la ventana de estatus							
Limitaciones: Se indicará el tiempo de ejecución de la simulación que se está ejecutando, y además se indicará si se ha producido un error.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	1.5	Nombre Requisito:	Instalador	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El instalador deberá realizar todos los tratamientos de datos y creación de archivos, necesarios para la correcta ejecución del programa							
Criterio de verificación: Una vez instalado el programa se ejecuta correctamente. Relacionado con el requisito 1.1							
Limitaciones: Solo debe funcionar en Windows Vista/7							
Tipo de requisito: No Funcional.							
Estado: Aprobado							

Nº	1.5.1	Nombre Requisito:	Incluir librerías	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El instalador deberá incluir las librerías necesarias para el correcto funcionamiento de la GUI conjuntamente con el simulador empleado (V-REP). Esto es, durante el proceso de instalación, se incluirán de forma correcta todas las librerías que requiera el simulador.							
Criterio de verificación: Al lanzar la aplicación, ésta se ejecutará sin ningún error derivado de la falta de inclusión de alguna librería, permitiendo al usuario hacer uso de todas las opciones que presenta la GUI. Los gráficos de la ventana del simulador se verán correctamente, pudiendo observar sin problemas el movimiento del robot dentro del laberinto, así como su interacción con el mismo.							
Limitaciones: Ninguna							
Tipo de requisito: No Funcional.							
Estado: Aprobado							

Nº	1.5.2	Nombre Requisito:	Copia de archivos necesarios	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Durante el proceso de instalación, el instalador será capaz de copiar los archivos necesarios que vaya a emplear la aplicación en cualquier momento de su ejecución.							
Criterio de verificación: Los archivos se copiarán correctamente y por lo tanto, no aparecerán mensajes de error en la instalación.							
Limitaciones: Ninguna							
Tipo de requisito: No Funcional.							
Estado: Aprobado							

Nº	1.5.3	Nombre Requisito:	Crear acceso directo	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El instalador de la aplicación creará un icono de acceso directo a la misma una vez completada la instalación.							
Criterio de verificación: En el directorio donde se haya guardado la aplicación podrá observarse la aparición de un icono representativo de la misma. Al hacer doble click sobre dicho icono se lanzará la aplicación.							
Limitaciones: Ninguna							
Tipo de requisito: No funcional.							
Estado: Aprobado							

Nº	1.6	Nombre Requisito:	Almacenar	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Esta acción se ejecutará cuando el usuario solicite guardar la simulación (1.4.2.6 Botón Guardar simulación). Se puede seleccionar la información a guardar (datos de simulación, programa del robot, laberintos, simulación completa) y se validarán las extensiones de los archivos dependiendo de lo que se pretende guardar. En caso de errores en la escritura a disco se notificará al usuario.							
Criterio de verificación: Después de realizar una tarea de almacenamiento los archivos deben encontrarse en la ubicación indicada y con las extensiones correspondientes.							
Limitaciones: No se debe permitir al usuario modificar los archivos creados por cada simulación.							
Tipo de requisito: No Funcional.							
Estado: Aprobado.							

Nº	1.6.1	Nombre Requisito:	Almacenar Datos	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Este proceso será lanzado por Almacenar (1.6) en el caso de que el usuario solicite almacenar los datos de la simulación o por Almacenar Simulaciones (1.6.4). Los datos se almacenarán en disco para cada paso de la simulación se guardará: Tiempo, coordenadas (x, y) del robot, velocidad del robot, estado de los sensores.							
Criterio de verificación: Después de realizar una tarea de almacenamiento, se guardará la información especificada para cada paso de la simulación.							
Limitaciones: La diversidad de datos que se pueden almacenar se ve afectada por: Robot utilizado (Kephera III), información disponible del motor físico (ODE o Bullet).							
Tipo de requisito: No Funcional.							
Estado: Aprobado							

Nº	1.6.2	Nombre Requisito:	Almacenar Laberinto	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Este proceso será lanzado por Almacenar (1.6) en el caso donde el usuario solicita almacenar el laberinto de la simulación o por Almacenar Simulaciones (1.6.4). El laberinto se almacenará en disco y contendrá la información necesaria para volver a dibujar el laberinto en la ventana de entorno (1.4.3.1) en futuras sesiones.							
Criterio de verificación: Después de realizar una tarea de almacenamiento, el archivo contendrá la información necesaria para volver a dibujar el laberinto en la ventana de entorno.							
Limitaciones: Cada archivo contendrá solo un laberinto.							
Tipo de requisito: No Funcional.							
Estado: Aprobado.							

Nº	1.6.3	Nombre Requisito:	Almacenar Programa Robot	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Este proceso será lanzado por Almacenar (1.6) en el caso donde el usuario solicita almacenar el programa del robot o por Almacenar Simulaciones (1.6.4). El programa se almacenará en disco.							
Criterio de verificación: Después de realizar una tarea de almacenamiento, el archivo contendrá el programa del robot.							
Limitaciones: Cada archivo contendrá solo un programa.							
Tipo de requisito: No Funcional.							
Estado: Aprobado.							

Nº	1.6.4	Nombre Requisito:	Almacenar Simulaciones	Fecha:	13/04/2012	Persona responsable:	Francisco Suárez
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Este proceso será lanzado por Almacenar (1.6) en el caso donde el usuario solicita almacenar las simulaciones. Los datos que se almacenarán serán los indicados en: Almacenar Datos (1.6.1), Almacenar Laberinto (1.6.2) y Almacenar Programa de Robot (1.6.3)							
Criterio de verificación: Después de realizar una tarea de almacenamiento los archivos contendrán la información correspondiente a la simulación.							
Limitaciones: Cada archivo contendrá sólo una simulación y, por lo tanto, unos datos, un laberinto y un código.							
Tipo de requisito: No Funcional.							
Estado: Aprobado.							

Nº	1.7	Nombre Requisito:	Tipos de usuario	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Baja	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: La aplicación dispondrá de dos modalidades de usuario: usuario básico y administrador, el cual tendrá privilegios.							
Criterio de verificación: Verificación de los requisitos 1.7.1 y 1.7.2							
Limitaciones: Sólo podrá haber un usuario conectado a la vez.							
Tipo de requisito: Funcional							
Estado: Opcional							

Nº	1.7.1	Nombre Requisito:	Usuario básico	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Baja	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: El usuario podrá cargar, y cerrar el laberinto. En caso de que el laberinto sea creado por él, también modificar, guardar y/o borrar. También podrá cargar, editar, modificar, guardar y borrar el código del robot.							
Criterio de verificación: Cuando el usuario acceda sin introducir ninguna clave, este accederá como usuario básico.							
Limitaciones: En ningún caso tendrá los privilegios del administrador.							
Tipo de requisito: Funcional							
Estado: Opcional							

Nº	1.7.2	Nombre Requisito:	Administrador	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Baja	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: Si el usuario es el identificado como administrador, se le permitirá la gestión del laberinto y del robot en todas sus características espaciales y de percepción.							
Criterio de verificación: Cuando el usuario acceda con cuenta de administrador, obtendrá los privilegios de éste.							
Limitaciones: Ninguna							
Tipo de requisito: Funcional							
Estado: Opcional							

3.2 Requisitos físicos

3.2.1 Simular Robot Kephhera III

Nº	2.1.1	Nombre Requisito:	Velocidad máxima del robot	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La velocidad máxima que podrá alcanzar el robot durante toda la simulación será de 0,5 m/s.							
Criterio de verificación: Vendrá impuesto internamente en las condiciones del programa.							
Limitaciones: Esta velocidad máxima estará limitada por el propio programa a 0,5 m/s. No será sobrepasada en ningún momento de la simulación.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.1.2	Nombre Requisito:	Peso del robot	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El peso del robot a tener en cuenta dentro del entorno de la simulación será de 690 g. Este peso representa fielmente el peso que podría presentar un robot Kephhera III en condiciones normales, incluyendo todos los elementos de sensado.							
Criterio de verificación: Vendrá impuesto internamente en las condiciones del programa.							
Limitaciones: El peso del robot a emplear será de 690 g; y en ningún caso será diferente, a no ser que se especifique de forma interna en el programa para probar posibles comportamientos del robot con la adición o sustracción de sensores, lo cual haría variar el peso del mismo.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.1.3	Nombre Requisito:	Dimensiones del robot	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Las dimensiones del robot Kephera III empleado en las distintas simulaciones, por construcción, serán de: 130 mm de diámetro y 70 mm de altura.							
Criterio de verificación: Se podrán verificar las correctas dimensiones del robot por comparativa visual con los distintos elementos del entorno, como las paredes del laberinto, cuya altura será especificada más adelante en el punto 2.3.1.							
Limitaciones: Las dimensiones del robot permanecerán constantes, no pudiéndose variar en ningún momento.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.1.4	Nombre Requisito:	Sensores empleados por el robot	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Los sensores que presentará el robot para el reconocimiento de su entorno serán: 8 sensores de infrarrojos y 5 sensores de ultrasonidos.							
Criterio de verificación: Se podrá verificar de forma visual durante la simulación el número de sensores que incorpora el robot.							
Limitaciones: El número máximo de sensores empleados será el especificado en la descripción del presente requisito.							
Tipo de requisito: Dominio.							
Estado: Aprobado							

3.2.2 Condiciones de la Tierra

Nº	2.2.1	Nombre Requisito:	Gravedad	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El valor de la gravedad será de $9,81 \text{ m/s}^2$.							
Criterio de verificación: Vendrá impuesto internamente en las condiciones del programa.							
Limitaciones: Este valor no podrá ser modificado, ya que el simulador pretende recrear las condiciones dentro de la órbita terrestre.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.2.2	Nombre Requisito:	Rozamiento	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El valor del coeficiente de rozamiento estimado para el suelo será de 0,5.							
Criterio de verificación: Vendrá impuesto internamente en las condiciones del programa.							
Limitaciones: Las derivadas del tipo de suelo en el que se pretenda realizar la simulación. En principio este tipo de suelo siempre será el mismo, por lo que el valor del coeficiente de rozamiento permanecerá constante.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

3.2.3 Condiciones del laberinto

Nº	2.3.1	Nombre Requisito:	Altura mínima de paredes	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La altura de todas las paredes que compondrán un laberinto será de 20 cm. Este requisito viene impuesto por el concurso <i>Cybertech</i> al que pretende dar solución la presente aplicación.							
Criterio de verificación: Se podrá verificar la altura de las paredes del laberinto por comparativa visual con los distintos elementos del entorno, como las dimensiones del robot, especificadas anteriormente.							
Limitaciones: Esta magnitud no podrá tomar valores menores o mayores por ser un requisito impuesto por el concurso.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.3.2	Nombre Requisito:	Distancia entre paredes	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La distancia entre las distintas paredes que compondrán un laberinto será de 30 cm. Este requisito viene impuesto por el concurso <i>Cybertech</i> al que pretende dar solución la presente aplicación.							
Criterio de verificación: Se podrá verificar la distancia entre paredes del laberinto por comparativa visual con los distintos elementos del entorno, como las dimensiones del robot o la altura de las paredes.							
Limitaciones: Esta magnitud no podrá tomar valores menores o mayores por ser un requisito impuesto por el concurso.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.3.3	Nombre Requisito:	Unión entre paredes y el suelo	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Las paredes estarán unidas entre ellas y con el suelo formando un ángulo de 90º. Este requisito viene impuesto por el concurso <i>Cybertech</i> al que pretende dar solución la presente aplicación.							
Criterio de verificación: Este requisito podrá ser verificado de forma visual por el usuario durante la simulación desde las distintas perspectivas que estarán a su disposición. Por ejemplo, seleccionando la perspectiva de vista en planta y vista frontal, se podrá comprobar perfectamente el ángulo recto formado por la unión entre paredes y de estas con el suelo, respectivamente.							
Limitaciones: La correcta observación de este requisito dependerá en mayor o menor medida de la calidad gráfica proporcionada por el simulador.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.3.4	Nombre Requisito:	Color de las paredes y el suelo	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El color de las superficies de las paredes y del suelo será blanco. Este requisito viene impuesto por el concurso <i>Cybertech</i> al que pretende dar solución la presente aplicación.							
Criterio de verificación: Este requisito podrá ser verificado de forma visual por el usuario durante la simulación.							
Limitaciones: La correcta observación de este requisito dependerá en mayor o menor medida de la calidad gráfica proporcionada por el simulador.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.3.5	Nombre Requisito:	Existencia de islas	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	Estab	Release:	1.0
Descripción: Una de las características más importantes del laberinto es la ausencia de islas en el mismo. Este requisito viene impuesto por el concurso <i>Cybertech</i> al que pretende dar solución la presente simulación.							
Criterio de verificación: Este requisito podrá ser verificado de forma visual por el usuario durante la simulación.							
Limitaciones: Ninguna							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

Nº	2.3.6	Nombre Requisito:	Entrada y salida del laberinto	Fecha:	13/04/2012	Persona responsable:	Carlos Sampedro
Prioridad:			Alta	Estabilidad del Documento	No Puede Cambiar	Release:	1.0
Descripción: El laberinto deberá tener una entrada al mismo y una salida del mismo. Este requisito viene impuesto por el concurso <i>Cybertech</i> al que pretende dar solución el presente simulador.							
Criterio de verificación: Este requisito podrá ser verificado de forma visual por el usuario durante la simulación.							
Limitaciones: Solo debe existir una salida y una entrada.							
Tipo de requisito: Dominio.							
Estado: Aprobado.							

3.3 Requisitos de la simulación

Nº	3.1	Nombre Requisito:	Velocidad	Fecha:	13/04/2012	Persona responsable:	Álvaro López
Prioridad :			Media	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: Modificación del parámetro de velocidad en la que se ejecuta la simulación. Pudiéndose incrementar o disminuir la velocidad respecto del tiempo real.							
Criterio de verificación: Verificación de los requisitos 3.1.1 y 3.1.2 .							
Limitaciones: Limitaciones de los requisitos 3.1.1 y 3.1.2 .							
Tipo de requisito: Funcional.							
Estado: Opcional							

Nº	3.1.1	Nombre Requisito:	Simular con velocidad cercana al tiempo real	Fecha:	13/04/2012	Persona responsable:	Álvaro López
Prioridad :			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: La simulación se podrá ejecutar de manera que el tiempo de simulación sea próximo al real.							
Criterio de verificación: Se realizara una simulación que tendrá un tiempo de resolución próximo a el tiempo real de salida del laberinto en la realidad.							
Limitaciones: Limitaciones introducidas por el ordenador sobre el que se ejecute la simulación.							
Tipo de requisito: Funcional.							
Estado: Opcional							

Nº	3.1.2	Nombre Requisito:	Variar velocidad de simulación	Fecha:	13/04/2012	Persona responsable:	Álvaro López
Prioridad :			Media	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: Se podrá modificar la velocidad en la cual se ejecuta la simulación, tanto al alza como a la baja respecto del tiempo real de ejecución.							
Criterio de verificación: La misma simulación se podrá ejecutar en diferentes velocidades, llegando al final de esta en diferentes tiempos.							
Limitaciones: Limitaciones introducidas por el ordenador sobre el que se ejecute la simulación.							
Tipo de requisito: Funcional.							
Estado: Opcional.							

Nº	3.2	Nombre Requisito:	Cámara	Fecha:	13/04/2012	Persona responsable:	Marie Andréé Destarac
Prioridad:			Alta	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: En el cuadro de simulación se puede seleccionar la “Cámara” correspondiente a una de las tres posibles vistas: vista superior, vista en perspectiva isométrica y vista en primera persona del robot. Además, se permite variar el zoom de la cámara seleccionada.							
Criterio de verificación: La vista y el zoom seleccionado se despliegan correctamente en el simulador.							
Limitaciones: Sólo puede seleccionarse a la vez una de las tres vistas .El zoom seleccionado tampoco puede ser modificado durante la simulación.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	3.2.1	Nombre Requisito:	Vista superior	Fecha:	13/04/2012	Persona responsable:	Marie Andréé Destarac
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: En el cuadro de simulación se mostrará la vista superior del laberinto seleccionado y del robot.							
Criterio de verificación: Al seleccionar la opción “Vista superior” de “Cámara”, se mostrará el laberinto y el robot en vista superior.							
Limitaciones: La perspectiva mostrada al seleccionar esta vista dependerá del zoom seleccionado por el usuario.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	3.2.2	Nombre Requisito:	Vista isométrica	Fecha:	13/04/2012	Persona responsable:	Marie Andréé Destarac
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: En el cuadro de simulación se mostrará la vista isométrica del laberinto seleccionado y del robot.							
Criterio de verificación: Al seleccionar la opción “Vista isométrica” de “Cámara”, se mostrará el laberinto y el robot en vista isométrica.							
Limitaciones: La perspectiva mostrada al seleccionar esta vista dependerá del zoom seleccionado por el usuario.							
Tipo de requisito: Funcional.							
Estado: Aprobada							

Nº	3.2.3	Nombre Requisito:	Vista en primera persona	Fecha:	13/04/2012	Persona responsable:	Marie Andréé Destarac
Prioridad:			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: En el cuadro de simulación se mostrará la vista en primera persona del laberinto seleccionado. Dicha vista se entiende como la que se percibiría al montar la cámara en el robot mismo.							
Criterio de verificación: Al seleccionar la opción “Vista en primera persona” de “Cámara”, se mostrará el laberinto visto desde el robot.							
Limitaciones: La perspectiva mostrada al seleccionar esta vista dependerá del zoom seleccionado por el usuario.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	3.2.4	Nombre Requisito:	Variar el zoom	Fecha:	13/04/2012	Persona responsable:	Marie Andréé Destarac
Prioridad:			Media	Estabilidad del Documento	Puede cambiar	Release:	1.0
Descripción: En el cuadro de simulación, en Cámara, podrá aumentarse o disminuirse el zoom de la cámara. Dicha aplicación funciona para cualquiera de las tres vistas disponibles.							
Criterio de verificación: Al aumentar el zoom con el botón correspondiente, la imagen del cuadro de simulación aumentará de tamaño en la perspectiva seleccionada. Al disminuir el zoom con el botón correspondiente, la imagen del cuadro de simulación disminuirá de tamaño en la perspectiva seleccionada.							
Limitaciones: El zoom tiene un rango limitado.							
Tipo de requisito: Funcional.							
Estado: Opcional							

Nº	3.3	Nombre Requisito:	Ubicación de objetos	Fecha:	13/04/2012	Persona responsable:	Álvaro López
Prioridad :			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Ubicación de los diferentes elementos dentro de la ventana de entorno (1.4.3.1), en el que se realice la simulación.							
Criterio de verificación: Verificación de los requisitos 3.3.1, 3.3.2 y 3.3.3							
Limitaciones: Limitaciones de los requisitos 3.3.1, 3.3.2 y 3.3.3							
Tipo de requisito: Funcional.							
Estado: Aprobado.							

Nº	3.3.1	Nombre Requisito	Situar inicialmente al robot en cualquier parte del laberinto.	Fecha:	13/04/2012	Persona responsable:	Álvaro López
Prioridad :			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: El robot se posicionará sobre el laberinto en un punto definido por el usuario.							
Criterio de verificación: El robot aparece sobre las coordenadas establecidas por el usuario. Además, estas se visualizarán en una ventana destinada a tal efecto.							
Limitaciones: El punto deberá estar dentro del laberinto.							
Tipo de requisito: Funcional.							
Estado: Aprobado							

Nº	3.3.2	Nombre Requisito:	Ubicar las paredes del laberinto a gusto del usuario	Fecha:	13/04/2012	Persona responsable:	Álvaro López
Prioridad :			Alta	Estabilidad del Documento	Estable	Release:	1.0
Descripción: Las paredes se posicionarán sobre el suelo en la ventana de entorno (1.4.3.1), definidas por dos puntos, introducidos por el usuario. Una vez se inicie la simulación el usuario no podrá introducir ni modificar las paredes existentes en el laberinto.							
Criterio de verificación: Las paredes aparecen sobre las coordenadas establecidas por el usuario. Además, estas se visualizarán en la ventana de entorno (1.4.3.1),							
Limitaciones: El punto deberá estar dentro del laberinto y cumplir los requisitos: 2.3.2, 2.3.3, 2.3.5 y 2.3.6							
Tipo de requisito: Funcional/Dominio							
Estado: Aprobado.							

Nº	3.3.3	Nombre Requisito:	Generar laberintos aleatoriamente previa solicitud del usuario	Fecha:	13/04/2012	Persona responsable:	Marta Dorado
Prioridad:			Media	Estabilidad del Documento	Puede Cambiar	Release:	1.0
Descripción: Existirá la posibilidad de crear laberintos de forma automática.							
Criterio de verificación: Creación de un laberinto en su totalidad de forma aleatoria y automática.							
Limitaciones: Ninguna							
Tipo de requisito: Funcional/Dominio							
Estado: Opcional (por implementar)							

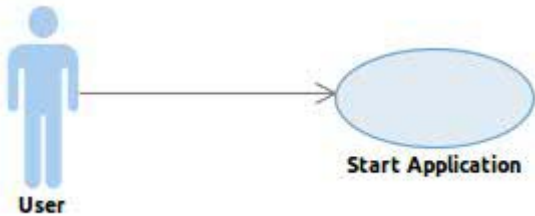
4. Diagramas del producto

4.1 Casos de uso

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

A continuación se detallan algunos de los casos de uso de la aplicación VCT:

1. INICIAR LA APLICACIÓN.

Actores:	Usuario
Descripción	El usuario inicia la aplicación haciendo doble click sobre el icono correspondiente, lanzándose la aplicación desde un archivo ".exe" sin necesidad de programas adicionales.
Precondiciones	La existencia del archivo ".exe" en un directorio accesible.
Postcondiciones	La aplicación se ha iniciado correctamente.
Curso Normal	1.0.- El usuario hace doble click sobre el archivo ".exe". 2.0.- La aplicación se lanza. 3.0.- Carga un entorno de inicio.
Excepciones	2.0.E.1. Hay errores en la iniciación de la aplicación. 3.0.E.1. Hay errores en la carga del entorno de inicio.
Prioridad	Alta.
Frecuencia de Uso	Media.
Requerimiento Especiales	La iniciación de esta aplicación no dependerá de programas adicionales.
 <pre>graph LR; User((User)) --> StartApplication([Start Application]);</pre>	

2. CARGAR LABERINTO PREDEFINIDO.

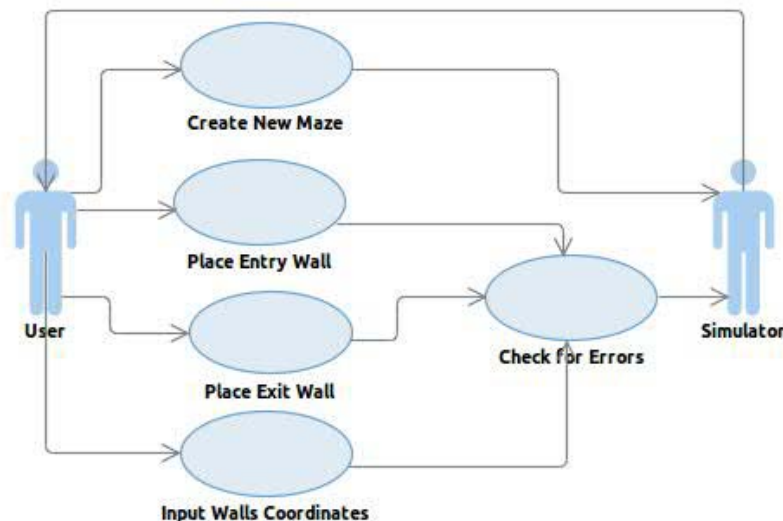
Actores:	Usuario; Simulador
Descripción	El usuario, a través de la <i>Interfaz de Usuario</i> , elige la opción de cargar un laberinto predefinido en el sistema; el cual se mostrará en la ventana correspondiente de la Interfaz.
Precondiciones	Los laberintos ya elaborados o predefinidos están disponibles en un directorio del sistema habilitado para tal efecto.
Postcondiciones	El laberinto seleccionado se muestra en la ventana de la <i>Interfaz de Usuario</i> .
Curso Normal	<p>1.0.- El usuario selecciona la opción de cargar un laberinto predefinido.</p> <p>2.0.- El usuario selecciona de un desplegable el laberinto sobre el que desea efectuar la simulación.</p> <p>3.0.- La <i>Interfaz de Usuario</i> llama al simulador para que cargue el laberinto seleccionado.</p> <p>4.0.- El Simulador dibuja cada uno de los componentes que conforman el laberinto.</p> <p>5.0.- El Simulador muestra a través de la <i>Interfaz de Usuario</i> el laberinto cargado.</p>
Curso Alternativo	No existe
Excepciones	<p>2.0.E.1. No existen laberintos predefinidos en el directorio.</p> <p>2.0.E.2. Hay errores en el proceso de carga del laberinto.</p>
Inclusiones	Laberinto predefinido
Prioridad	Media.
Frecuencia de Uso	Podría ser casi continuo
Notas y documentos	Fichero con laberinto predefinido

```

graph LR
    User((User)) --> SelectMaze([Select Maze])
    SelectMaze --> ShowMaze([Show Maze])
    ShowMaze --> LoadMaze([Load Maze])
    LoadMaze --> SystemFiles((System Files))
    SystemFiles --> User
  
```

3. CREAR LABERINTO NUEVO.

Actores:	Usuario normal; Simulador
Descripción	El usuario, a través de la <i>Interfaz de Usuario</i> , elige la opción de cargar un laberinto nuevo. Para ello, el usuario deberá ir dibujando las líneas que corresponden a las paredes en la retícula dispuesta para este fin en la interfaz de usuario. Por ultimo presiona “actualizar” para cargar las paredes al simulador.
Precondiciones	El usuario conoce las reglas en la introducción de las paredes.
Postcondiciones	Las paredes introducidas se irán visualizando en la <i>Interfaz de Usuario</i> hasta componer el laberinto una vez el usuario presione “actualizar”.
Curso Normal	1.0.- El usuario habilita la opción de cargar un laberinto nuevo. 2.0.- El usuario introduce dibuja paredes en la retícula. 3.0.- El usuario selecciona la opción de "Actualizar". 4.0.- La <i>Interfaz de Usuario</i> llama al simulador para que cargue una pared en la posición seleccionada por el usuario. 5.0.- El Simulador dibuja la pared en las coordenadas seleccionadas. 6.0.- El Simulador muestra a través de la <i>Interfaz de Usuario</i> la pared.
Curso Alternativo	2.1.- El usuario selecciona crear una pared de forma aleatoria.
Excepciones	2.0.E.1. La posición introducida por el usuario para fijar la pared en el laberinto no son válidas o están fuera de rango. 4.0.E.1. Hay errores en el proceso de carga de la pared.
Prioridad	Media
Frecuencia de Uso	Podría ser casi continuo



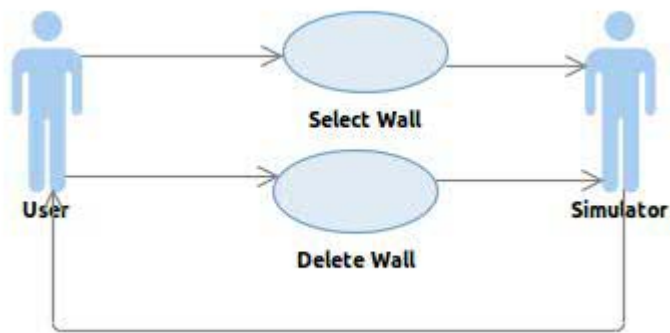
4. CREAR LABERINTO ALEATORIO.

Actores:	Usuario normal; Simulador
Descripción	El usuario, a través de la <i>Interfaz de Usuario</i> , elige la opción de cargar un laberinto nuevo aleatorio.
Precondiciones	
Postcondiciones	El laberinto generado aleatoriamente se visualizará en la <i>Interfaz de Usuario</i>
Curso Normal	1.0.- El usuario habilita la opción de cargar un laberinto nuevo aleatorio. 2.0.- La <i>Interfaz de Usuario</i> llama al simulador para que cargue el laberinto creado aleatoriamente. 3.0.- El Simulador muestra a través de la <i>Interfaz de Usuario</i> el laberinto generado.
Excepciones	1.0.E.1. Las coordenadas generadas para posicionar la pared en el laberinto no son válidas o están fuera de rango. 1.0.E.2. Hay errores en el proceso de carga de la pared.
Prioridad	Media
Frecuencia de Uso	Podría ser casi continuo
<pre> graph LR User((User)) --> Create(Create Random Maze) Create --> Execute(Execute Random Function) Execute --> Simulator((Simulator)) Simulator --> User </pre>	

5. EDITAR PARED.

Actores:	Usuario normal, Simulador
Descripción	El usuario desea modificar una de las paredes existentes en el laberinto
Precondiciones	Existe un directorio creado para almacenar los datos de las simulaciones que tiene simulaciones guardadas, incluyendo: robot, laberinto y suelo
Postcondiciones	Existen en el programa un laberinto creado o en creación con al menos una pared.
Curso Normal	<p>1.0.- El usuario agrega o quita paredes de la retícula.</p> <p>2.0.- Presiona la opción “ Actualizar”</p> <p>3.0.- La <i>Interfaz de Usuario</i> comunica al <i>Simulador</i> el nuevo estado del laberinto.</p> <p>4.0.- La <i>Interfaz de Usuario</i> muestra el nuevo estado del laberinto.</p>
Excepciones	<p>2.0. E.1. No se muestra la posición de la primera pared o sí se hace, pero de otra pared</p> <p>4.0. E.1. Se produce un error en la comunicación <i>Interfaz de Usuario/Simulador</i></p> <p>6.0. E.1. No se muestra la posición de la primera pared o sí se hace, pero de otra pared</p> <p>8.0. E.1. Se produce un error en la comunicación <i>Interfaz de Usuario/Simulador</i></p>
Prioridad	Alta
Frecuencia de Uso	Podría ser casi continuo
<pre> sequenceDiagram actor User actor Simulator User->>UC1: Select Wall activate UC1 UC1->>UC2: Change Wall Coordinates activate UC2 UC2->>UC3: Check for Errors activate UC3 UC3->>Simulator deactivate UC3 Simulator->>User deactivate UC1 </pre> <p>The diagram illustrates the process of editing a wall. It involves three use cases: 'Select Wall', 'Change Wall Coordinates', and 'Check for Errors'. The process starts with the 'User' actor initiating the 'Select Wall' use case. This leads to the 'Change Wall Coordinates' use case, which then leads to the 'Check for Errors' use case. Finally, the 'Check for Errors' use case sends data to the 'Simulator' actor, who then sends data back to the 'User' actor, completing the cycle.</p>	

6. BORRAR PARED.

Actores:	Usuario , Simulador
Descripción	El usuario desea borrar una de las paredes que forman parte del laberinto
Precondiciones	Existen en el programa un laberinto creado o en creación con al menos una pared.
Postcondiciones	La última pared introducida es eliminada del laberinto y por lo tanto es borrada de la Interfaz de Usuario.
Curso Normal	1.0.- El usuario quita paredes de la retícula. 2.0.- Presiona la opción “ Actualizar” 3.0.- La <i>Interfaz de Usuario</i> comunica al <i>Simulador</i> el nuevo estado del laberinto. 4.0.- La <i>Interfaz de Usuario</i> muestra el nuevo estado del laberinto.
Curso Alternativo	El usuario desea la edición de la pared.
Excepciones	2.0. E.1. Existe un error a la hora de borrar la pared. 3.0. E.1. Se produce un error en la comunicación Interfaz de Usuario/Simulador
Prioridad	Alta
Frecuencia de Uso	Podría ser casi continuo
 <pre> sequenceDiagram actor User actor Simulator User->>Select Wall Simulator->>User User->>Delete Wall Simulator->>User </pre> <p>The diagram illustrates the interaction between a User and a Simulator for the 'BORRAR PARED' use case. It features two actors, 'User' and 'Simulator', represented by blue stick figures. There are two use cases, 'Select Wall' and 'Delete Wall', represented by light blue ovals. The flow of the process is as follows: 1. The User initiates the process by sending a message to the 'Select Wall' use case. 2. The Simulator then sends a message back to the User. 3. The User sends a message to the 'Delete Wall' use case. 4. Finally, the Simulator sends a message back to the User. This sequence represents the user's interaction with the system to remove a wall.</p>	

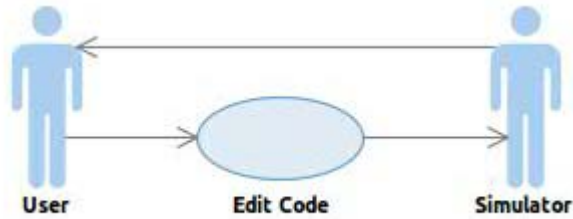
7. POSICIONAR ROBOT EN EL LABERINTO.

Actores:	Usuario; Simulador
Descripción	El usuario, a través de la <i>Interfaz de Usuario</i> , arrastra en la retícula el punto que corresponde al robot hasta el lugar deseado.
Precondiciones	El suelo debe haber sido creado previamente.
Postcondiciones	El robot se visualizará en la posición fijada por el usuario en la <i>Interfaz de Usuario</i> .
Curso Normal	<p>1.0.- El usuario arrastra el robot hasta la posición deseada con el ratón.</p> <p>2.0.- El usuario selecciona la opción de "actualizar".</p> <p>3.0.- La <i>Interfaz de Usuario</i> llama al simulador para que cargue el robot en las coordenadas seleccionadas por el usuario.</p> <p>4.0.- El Simulador dibuja el robot en las coordenadas seleccionadas.</p> <p>5.0.- El Simulador muestra a través de la <i>Interfaz de Usuario</i> el robot cargado.</p>
Curso Alternativo	No existe
Excepciones	<p>1.0.E.1. Las posición introducida por el usuario para posicionar el robot en el laberinto no son válidas o están fuera de rango.</p> <p>3.0.E.1. Hay errores en el proceso de carga del robot.</p>
Prioridad	Alta
Frecuencia de Uso	Podría ser casi continuo
<pre> graph LR User((User)) --> Input([Input Robot Coordinates]) Input --> Check([Check for Errors]) Check --> Simulator((Simulator)) Simulator --> Locate([Locate Robot]) Locate --> User Simulator --> User </pre>	

8. CARGAR CÓDIGO DEL ROBOT.

Actores:	Usuario
Descripción	El usuario o administrador selecciona con el click derecho sobre el robot la opción de cargar el código del robot en la interfaz, este código se carga correctamente y se puede simular su comportamiento.
Precondiciones	El código del robot se encuentra en el directorio apropiado y guardado en el formato valido.
Postcondiciones	El código del robot se ha cargado de manera correcta y se puede simular el robot con este código.
Curso Normal	1.0.- El usuario hace click derecho sobre el robot y selecciona la opción de cargar un código de robot ya guardado. 2.0.-Selecciona el archivo a cargar. 3.0.- La consola de edición de código se abre en caso de estar cerrada. 4.0.- La consola de edición de código carga con el archivo guardado.
Curso Alternativo	3.1- Si la consola está abierta pasa directamente a 4.0
Excepciones	2.0.E.1. Hay errores en la selección del archivo. 2.0.E.2. El archivo no se encuentra en el directorio válido. 2.0.E.3. El archivo no está en el formato válido. 3.0.E.1. Hay errores en la iniciación del editor de código. 4.0.E.1. Hay errores en la carga del editor de código.
Prioridad	Media.
Frecuencia de Uso	Podría ser casi continuo .
Notas y documentos	Archivos con el código del robot guardado en directorio y formato valido.
<pre> graph LR User((User)) --> SelectCodeFile((Select Code File)) SelectCodeFile --> SystemFiles((System Files)) SystemFiles --> LoadCode((Load Code)) LoadCode --> ShowCode((Show Code)) ShowCode --> User </pre> <p>The diagram illustrates the process of loading robot code. It features three actors: 'User' (represented by a stick figure on the left), 'System Files' (represented by a stick figure on the right), and two use cases: 'Select Code File' (top center) and 'Show Code' (bottom left). The flow is as follows: The 'User' initiates the process by selecting a code file. This action leads to the 'Select Code File' use case, which then interacts with 'System Files'. From 'System Files', the process moves to the 'Load Code' use case, which then leads to the 'Show Code' use case. Finally, 'Show Code' returns the information to the 'User'.</p>	

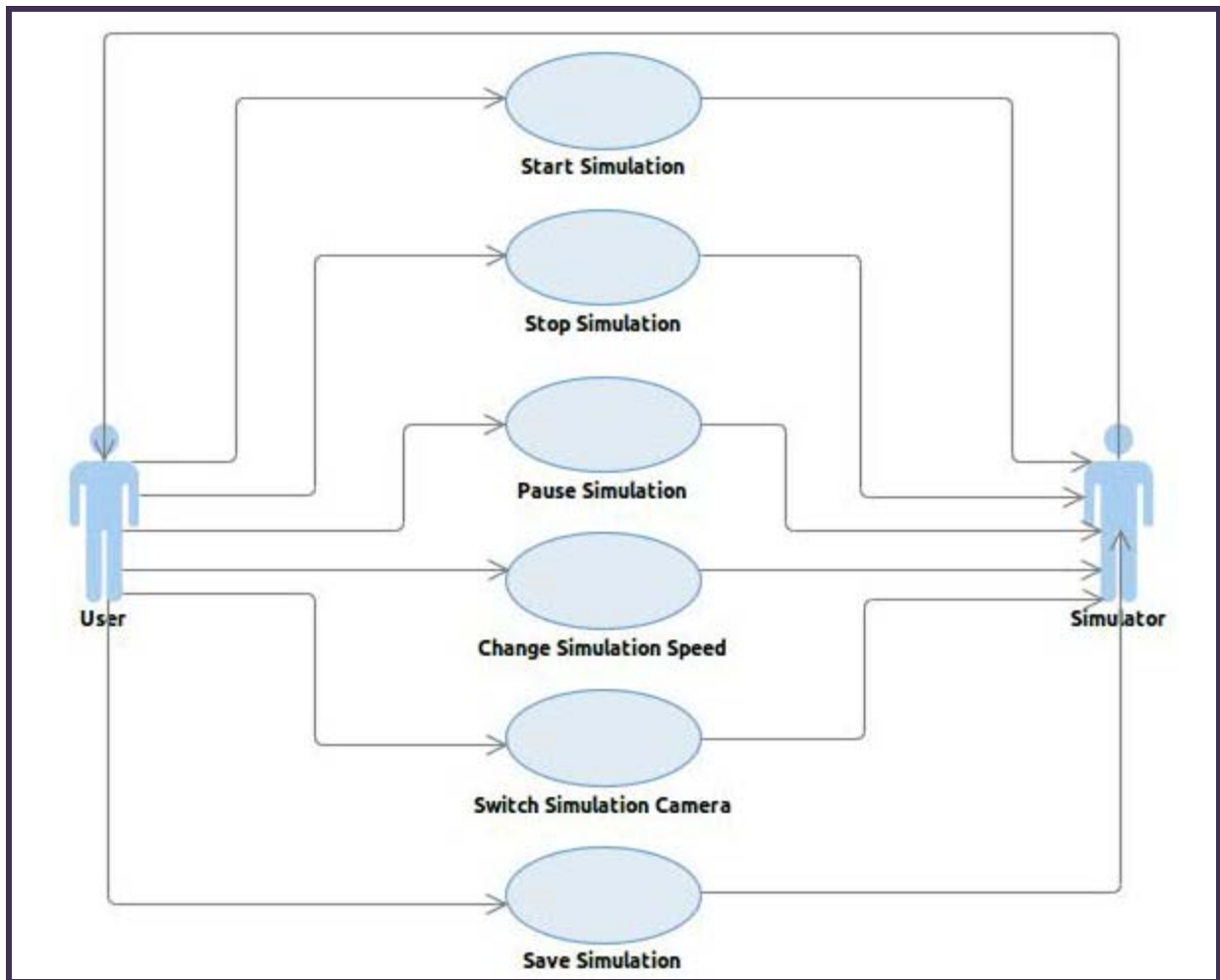
9. EDITAR CÓDIGO DEL ROBOT.

Actores:	Usuario; Simulador
Descripción	El usuario selecciona la opción de mostrar el código, en lenguaje "lua", actual del robot. En una ventana de la <i>Interfaz de Usuario</i> habilitada para ello se muestra el código actual del robot. El usuario podrá editar dicho código para probar sus algoritmos.
Precondiciones	El usuario conoce el lenguaje de programación "lua".
Postcondiciones	Se mostrará en la ventana de edición de código, el código del robot.
Curso Normal	<p>1.0.- El usuario presiona botón derecho del ratón y se selecciona la opción de " Editar el código actual del robot".</p> <p>2.0.- En la <i>Interfaz de Usuario</i> se muestra el código en lenguaje "lua" del robot.</p> <p>3.0.- El usuario edita el código.</p> <p>4.0.- La <i>Interfaz de Usuario</i> llama al Simulador para ejecutar dicho código.</p> <p>5.0.- El Simulador, a través de la <i>Interfaz de Usuario</i>, muestra el resultado de la ejecución.</p>
Curso Alternativo	4.1.- El usuario desea ejecutar paso a paso el código.
Excepciones	<p>3.0.E.1. El usuario introduce una sentencia de código incorrecta.</p> <p>4.0.E.1. Hay errores en el proceso de ejecución.</p>
Frecuencia de Uso	Podría ser casi continuo
 <pre> graph LR User((User)) --> EditCode([Edit Code]) EditCode --> Simulator((Simulator)) Simulator --> User </pre>	

10. INTERACCIÓN CON EL SIMULADOR.

Actores:	Usuario; Simulador
Descripción	El usuario selecciona dentro de la simulación la opción de iniciar, detener, pausar, cambiar la velocidad, cambiar la cámara o guardar la simulación, estando el robot en la posición deseada sobre el laberinto.
Precondiciones	El usuario ha cargado o creado un laberinto y ha posicionado el robot sobre el mismo, en las coordenadas deseadas.
Postcondiciones	<p>Si el usuario selecciona "Iniciar": Se irá mostrando en la ventana de simulación el movimiento del robot en el interior del laberinto.</p> <p>Si el usuario selecciona "Mover cámara": Los elementos que se encontraban en la <i>Interfaz de Usuario</i> han cambiado su perspectiva</p> <p>Si el usuario selecciona "Detener simulación": La simulación se ha detenido y no se podrá reanudar.</p> <p>Si el usuario selecciona "Pausar simulación": La simulación se ha pausado de manera correcta pudiéndose reiniciar desde ese instante.</p> <p>Si el usuario selecciona "Modificar velocidad de simulación": Pulsando botón "Faster" para aumentar la velocidad y "Slower" para disminuir la velocidad</p> <p>Si el usuario selecciona "Guardar simulación": Los datos de la simulación realizada serán guardados en el directorio habilitado para el almacenamiento de dichas simulaciones.</p>
Curso Normal	<p>Si el usuario selecciona "Iniciar": 1.0.- El Simulador muestra, a través de la <i>Interfaz de Usuario</i>, el movimiento del robot en el interior del laberinto.</p> <p>Si el usuario selecciona "Mover cámara": 1.0.- El usuario selecciona el cambio a una vista. 2.0.- La <i>Interfaz de Usuario</i> comunica al <i>Simulador</i> el cambio de cámara. 3.0.- La <i>Interfaz de Usuario</i> realiza el cambio de cámara</p> <p>Si el usuario selecciona "Detener simulación": 1.0.- El usuario tiene la simulación ejecutándose. 2.0.- El usuario pulsa el botón de detener simulación. 3.0.- La simulación se detiene.</p> <p>Si el usuario selecciona "Pausar simulación": 1.0.- El usuario pulsa el botón de pausar simulación. 2.0.- El usuario pulsa el botón de iniciar simulación.</p>

	<p>3.0.- La simulación se reinicia en el punto que se pauso. 4.0.- Se vuelve al punto 1.0.</p> <p>Si el usuario selecciona “Modificar velocidad de simulación”: 1.0.- El usuario presiona “Slower” o “Faster”. 2.0.- La <i>Interfaz de Usuario</i> comunica al <i>Simulador</i> el nuevo valor de la velocidad de simulación.</p> <p>Si el usuario selecciona “Guardar simulación”: 1.0.- El usuario selecciona la opción de "guardar simulación". 2.0.- El sistema guarda los ficheros que contienen los datos de la simulación en el directorio habilitado</p>
Excepciones	<p>Si el usuario selecciona “Iniciar”: 1.0. E.1. Hay errores en el proceso de simulación.</p> <p>Si el usuario selecciona “Mover cámara”: 1.0. E.1. Existe algún error en la comunicación <i>Interfaz de Usuario/Simulador</i> 2.0. E.1. La cámara no se cambia. 3.0. E.2. Se cambia a una vista que no es la deseada.</p> <p>Si el usuario selecciona “Detener simulación”: 1.0. E.1. Hay errores en la iniciación de la aplicación. 2.0. E.1. Hay errores en la carga del entorno de inicio.</p> <p>Si el usuario selecciona “Pausar simulación”: 3.0. E.1.-Hay errores en la reiniciación de la simulación.</p> <p>Si el usuario selecciona “Modificar velocidad de simulación”: 1.0. E.1. La <i>Interfaz de Usuario</i> desconoce a qué punto predefinido dirigirse al colocarlo el usuario el curso justo entre medias de dos posiciones predefinidas. 2.0. E.1 La comunicación <i>Interfaz de Usuario/Simulador</i> es errónea.</p> <p>Si el usuario selecciona “Guardar simulación”: 1.0.- El usuario selecciona la opción de "guardar simulación". 2.0.- El sistema guarda los ficheros que contienen los datos de la simulación en el directorio habilitado.</p>
Prioridad	Media, a excepción de modificar velocidad e iniciar que tienen prioridad alta.
Frecuencia de Uso	Podría ser casi continuo, a excepción de modificar velocidad que es esporádica.



11. CARGAR SIMULACIÓN.

Actores:	Usuario
Descripción	El usuario desea cargar una simulación ya sea para simularla de nuevo o para realizar cambios en ella
Precondiciones	Existe un directorio creado para almacenar los datos de las simulaciones que tiene simulaciones guardadas, incluyendo: robot, laberinto y suelo
Postcondiciones	Todos los elementos de la simulación son configurados en el sistema de la forma especificada en el archivo
Curso Normal	<p>1.0.- El usuario selecciona la opción de "cargar simulación".</p> <p>2.0.- El sistema carga los ficheros que contienen los datos de la simulación desde el directorio habilitado.</p> <p>3.0.- La <i>Interfaz de Usuario</i> comunica al <i>Simulador</i> el estado de la simulación.</p> <p>4.0.- La <i>Interfaz de Usuario</i> muestra el estado de la simulación</p>
Excepciones	<p>2.0. E.1. No se encuentra el directorio correspondiente.</p> <p>2.0. E.2. No se encuentra el archivo correspondiente</p> <p>2.0. E.3. El archivo no tiene el formato o los datos deseados.</p> <p>2.0. E.4. Los archivos no se cargan correctamente.</p> <p>3.0. E.1 La comunicación <i>Interfaz de Usuario/Simulador</i> es errónea.</p>
Inclusiones	Archivo con el formato deseado
Prioridad	Media
Frecuencia de Uso	Podría ser continuo
<pre> graph LR User((User)) -- "Select Simulation" --> SysFiles((System Files)) SysFiles -- "Load Simulation" --> User SysFiles -- "Show Simulation" --> User </pre>	

12. GUARDAR CÓDIGO.

Actores:	Usuario; Simulador
Descripción	Una vez finalizada la simulación, el usuario selecciona la opción de "guardar código".
Precondiciones	El usuario ha ingresado el código. Existe el directorio creado para almacenar el código.
Postcondiciones	El código realizado será guardado en el directorio habilitado para el almacenamiento de dichos códigos.
Curso Normal	1.0.- El usuario selecciona la opción de "guardar código". 2.0.-El usuario introduce el nombre del archivo de código 3.0.- El sistema guarda el fichero que contiene el código en el directorio habilitado.
Excepciones	3.0.E.1. No se encuentra el directorio correspondiente.
Prioridad	Media
Frecuencia de Uso	Podría ser continuo

```

sequenceDiagram
    actor User
    actor Simulator
    usecase UC1 as Select Code File
    usecase UC2 as Enter Code Name
    usecase UC3 as Save Code
    User->>UC1
    UC1->>Simulator
    User->>UC2
    UC2->>Simulator
    User->>UC3
    UC3->>Simulator
    Simulator->>User
    
```

The diagram illustrates the interaction between a User and a Simulator for the 'GUARDAR CÓDIGO' (Save Code) process. It features three use cases: 'Select Code File', 'Enter Code Name', and 'Save Code'. The User initiates the process by selecting a code file, entering a code name, and saving the code. Each of these actions is then processed by the Simulator, which finally returns control to the User.

13. GUARDAR LABERINTO.

Actores:	Usuario; Simulador
Descripción	Una vez finalizada la simulación, el usuario selecciona la opción de "guardar laberinto".
Precondiciones	El usuario ha creado/modificado/usado un laberinto. Existe el directorio creado para almacenar el laberinto.
Postcondiciones	El laberinto será guardado en el directorio habilitado para el almacenamiento de dichos laberintos.
Curso Normal	1.0.- El usuario selecciona la opción de "guardar laberinto". 2.0.-El usuario introduce el nombre del archivo de laberinto 3.0.- El sistema guarda el fichero que contiene el laberinto en el directorio habilitado.
Excepciones	3.0.E.1. No se encuentra el directorio correspondiente.
Prioridad	Media
Frecuencia de Uso	Podría ser continuo
<pre> sequenceDiagram actor User actor Simulator User->>UC1: Select Maze File UC1->>Simulator User->>UC2: Enter Maze Name UC2->>Simulator User->>UC3: Save Maze UC3->>Simulator Simulator->>User </pre> <p>The diagram illustrates the 'GUARDAR LABERINTO' process. It features two actors: 'User' and 'Simulator'. The process consists of three use cases: 'Select Maze File', 'Enter Maze Name', and 'Save Maze'. The flow starts with the User initiating the 'Select Maze File' use case, which then transitions to the Simulator. This pattern repeats for 'Enter Maze Name' and 'Save Maze'. Finally, the Simulator returns control to the User, completing the cycle.</p>	

14. CREAR NUEVA SIMULACIÓN.

Actores:	Usuario; Simulador
Descripción	El usuario, a través de la <i>Interfaz de Usuario</i> , elige la opción de crear nueva simulación.
Precondiciones	Ninguna
Postcondiciones	Se visualizará la simulación en la <i>Interfaz de Usuario</i>
Curso Normal	1.0.- El usuario habilita la opción de crear nueva simulación. 2.0.- El Simulador muestra a través de la <i>Interfaz de Usuario</i> el entorno de simulación y trabajo
Excepciones	1.0.E.1. Hay errores en el proceso de habilitar nueva simulación
Prioridad	Alta
Frecuencia de Uso	Podría ser continuo
<pre>graph LR; User((User)) --> CreateNewSimulation((Create New Simulation)); CreateNewSimulation --> Simulator((Simulator)); Simulator --> User;</pre> <p>The diagram illustrates the 'Create New Simulation' use case. It features two actors, 'User' and 'Simulator', represented by blue stick figures. A central light blue oval represents the use case 'Create New Simulation'. An arrow points from the 'User' actor to the use case, and another arrow points from the use case to the 'Simulator' actor. A return arrow points from the 'Simulator' actor back to the 'User' actor, indicating a feedback loop.</p>	

4.2. Interfaz de usuario

La figura 1 muestra la apariencia de la interfaz de usuario y las diferentes partes integrantes.

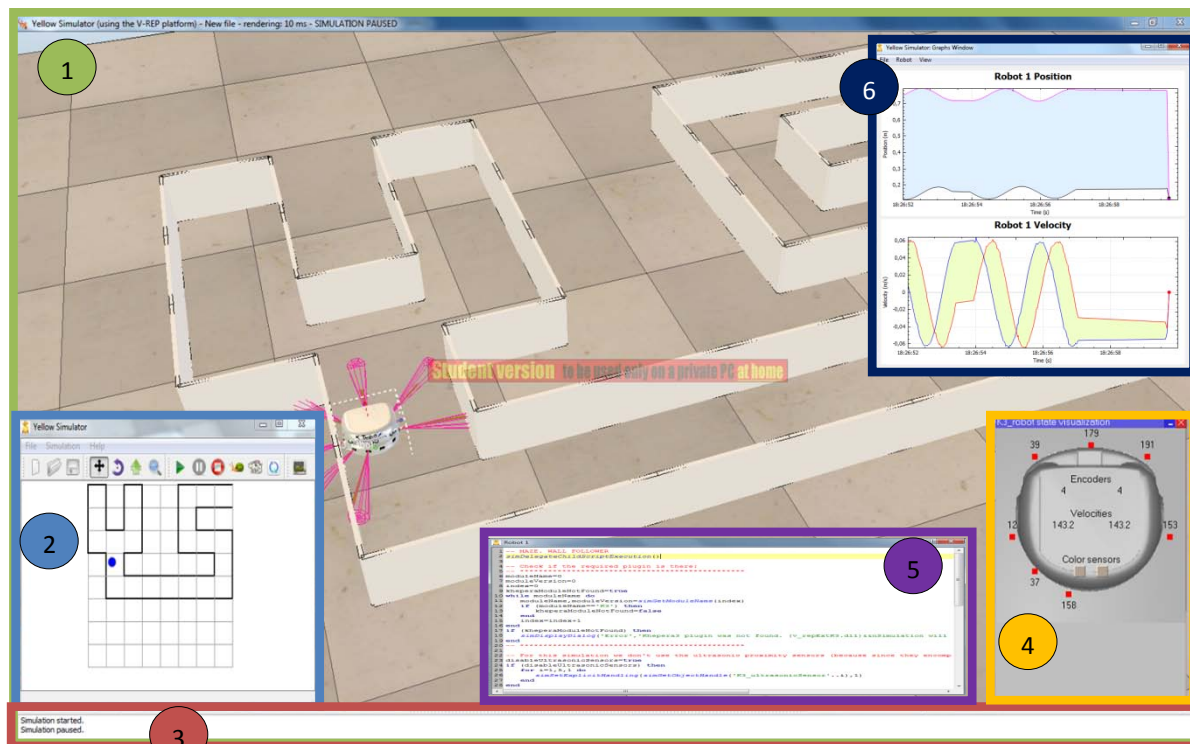


Figura 1. Interfaz de usuario del simulador

Definiremos los elementos integrantes por orden de aparición:

- 1.-Ventana de simulación: Permite visualizar el laberinto y la interacción del robot con éste durante la prueba del código introducido.
- 2.-Ventana de controles: Es la ventana principal del software desde la cual se accede a la mayoría de las funcionalidades de la aplicación. En ella es posible controlar la simulación, generar los laberintos, salvar archivos, entre otras funciones.
- 3.-Barra de estado: En ella se presentan los errores que se produzcan durante la compilación del código, así como el estado de la simulación.
- 4.-Ventana del estado del robot: En esta ventana se muestra el estado de los sensores del KIII durante la realización de la simulación y esa información se actualiza en tiempo real.
- 5.-Ventana de código: En esta ventana se introduce y edita el código de control a ejecutar en el robot durante la simulación.
- 6.-Ventana de información: En ella se puede observar gráficamente los valores de velocidad y posición del robot en tiempo real.

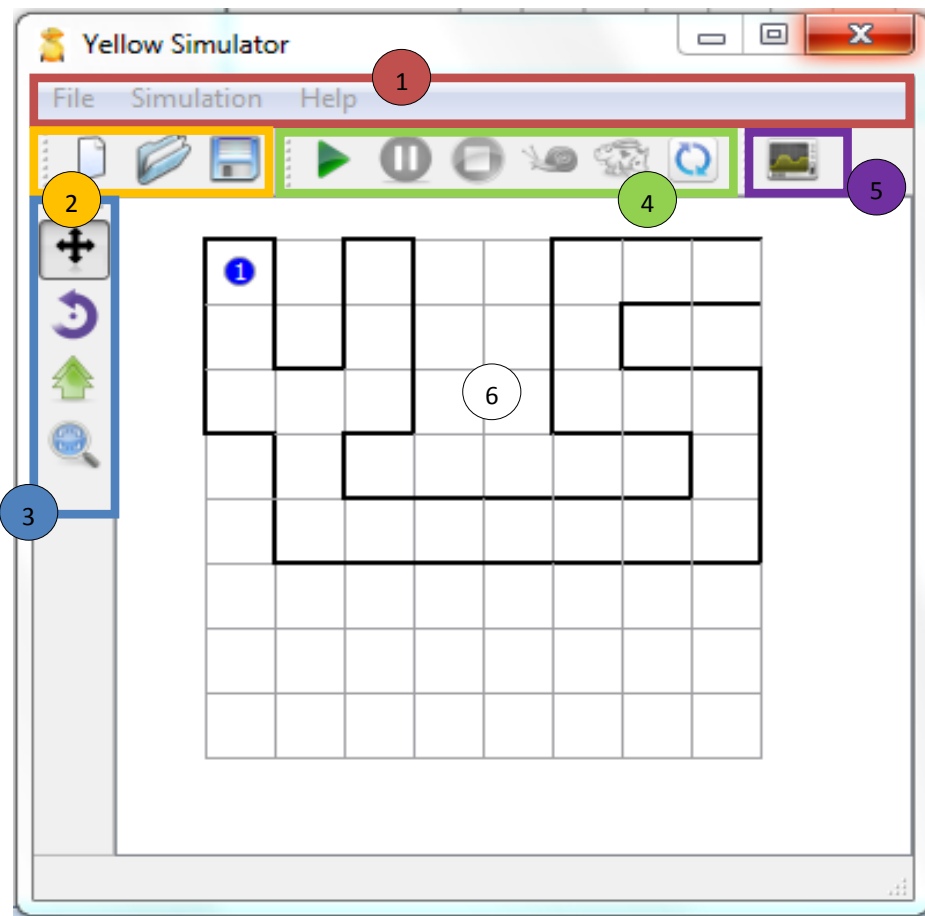


Figura 2. Ventana de controles de la interfaz de usuario

Definiremos los elementos integrantes por orden de aparición:

- 1.-Barra de menús: se compone de tres pestañas desplegables, que son File, Simulation y Help. Cada una permite acceder a diferentes funcionalidades del sistema.
- 2.-Barra de archivos: Permite crear, abrir y guardar los diferentes archivos del sistema.
- 3.-Barra de control de cámara: Permite ajustar la vista y el zoom.
- 4.-Barra de control de simulación: Permite controlar diferentes aspectos de la simulación, como la velocidad y el inicio de ésta.
- 5.-Barra de gráficas: Da acceso a la ventana con las gráficas del estado del robot.
- 6.-Ventana de edición del laberinto: Permite la edición del laberinto que el usuario desea crear y la colocación del robot en la posición deseada.