

# **Sistema de navegación de un robot basado en el reconocimiento visual de patrones en tiempo real**

## **Resumen de la Memoria**



### **Autor**

Rubén González del Pozo

### **Tutores**

Vidal Moreno Rodilla

Roberto Therón Sánchez

Departamento de Informática y Automática  
Universidad de Salamanca



## 1.Introducción

La Robótica es una rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas repetitivas o peligrosas para el ser humano.

A mediados del siglo XX, con la introducción de los microprocesadores, los modernos ordenadores han ofrecido un *cerebro* a los *músculos* de los robots mecánicos primitivos, cuya fusión hizo posible la aparición de los robots autónomos.

El marco de este Proyecto de Fin de Carrera se centra en la problemática de la navegación autónoma en el campo de los robots móviles.

## 2.Objetivos Alcanzados

El objetivo central de este proyecto fue el desarrollo del sistema de navegación de un robot móvil mediante un control visual, a través de las imágenes tomadas por la cámara montada en él, y los datos tomados por los sonares del propio robot. Este objetivo global ha sido alcanzado en sus principales aspectos.



Figura 1 – Robot AmigoBot con el *Mac mini* y la cámara web instalados

Para ello se propusieron dos principios de funcionamiento: un principio reactivo, centrado en la evasión de los obstáculos encontrados en el medio durante el movimiento del robot y otro reflexivo, en el que el robot móvil presenta un comportamiento determinado en función del estímulo visual recibido.

Así, a partir de este objetivo central surgieron una serie de metas destinadas a cumplir el objetivo primario.

En primer lugar, con el fin de obtener los datos procedentes del robot se pusieron en marcha una serie de mecanismos de gestión de dispositivos, para lo que se utilizó el servidor multirobot Player como medio de interconexión.

Como sistema de visión para la navegación se utilizó una cámara web junto con técnicas de Realidad Aumentada para la extracción de datos de las mismas.

Para realizar las tareas de navegación, se hizo una optimización del sistema creando múltiples procesos paralelos y un mecanismo de intercambio de información entre los mismos.

Como punto final, se desarrolló una interfaz gráfica de usuario intuitiva y fácil de manejar para la supervisión del movimiento autónomo del robot, de las imágenes tomadas y de los valores obtenidos por los sensores de dicho robot.

### **3. Diseño del Sistema**

El diseño del sistema es el proceso que modela el dominio de la solución, reexaminando las clases del dominio del problema, refinándolas, extendiéndolas y organizándolas para mejorar su reutilización.

Para el diseño de este sistema, se creó una arquitectura con tres paquetes, cada uno de los cuales se encarga de gestionar un subsistema distinto en los que se dividió el sistema completo.

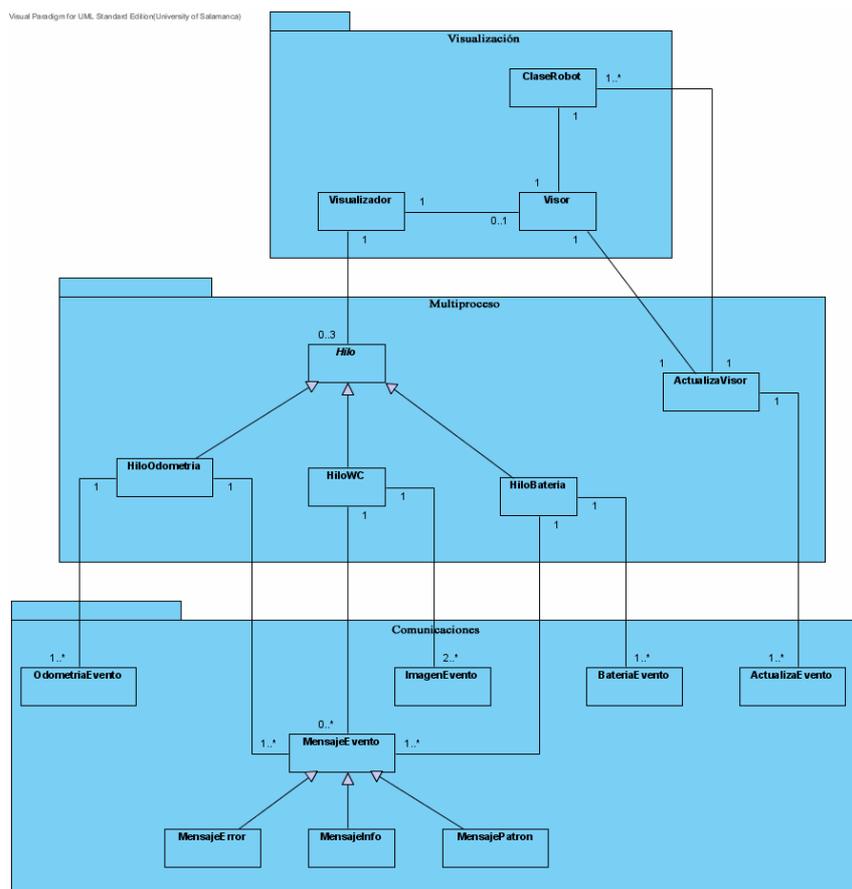


Figura 2 – Diagrama de clases del diseño

En la Figura 2 se puede observar la división en subsistemas y las relaciones entre los elementos de modelado pertenecientes a cada uno de ellos.

- El subsistema Comunicaciones es el encargado de gestionar el paso de mensajes e información entre los distintos objetos del sistema.
- El subsistema Multiproceso es el encargado de la creación de los distintos hilos de procesamiento destinados a la ejecución simultánea de las tareas de adquisición de imágenes y reconocimiento de patrones en ellas, movimiento autónomo del robot y toma de datos tanto odométricos y tomados por los sonares como del estado de la batería. También realiza la actualización de la representación de estos datos.
- El subsistema Visualización se ocupa de la representación visual de los datos obtenidos por los distintos objetos del sistema.

## 4.Aspectos Relevantes

Para la consecución de los objetivos propuestos al inicio del proyecto, se diseñó la aplicación atendiendo al esquema que se muestra a continuación (Figura 3):

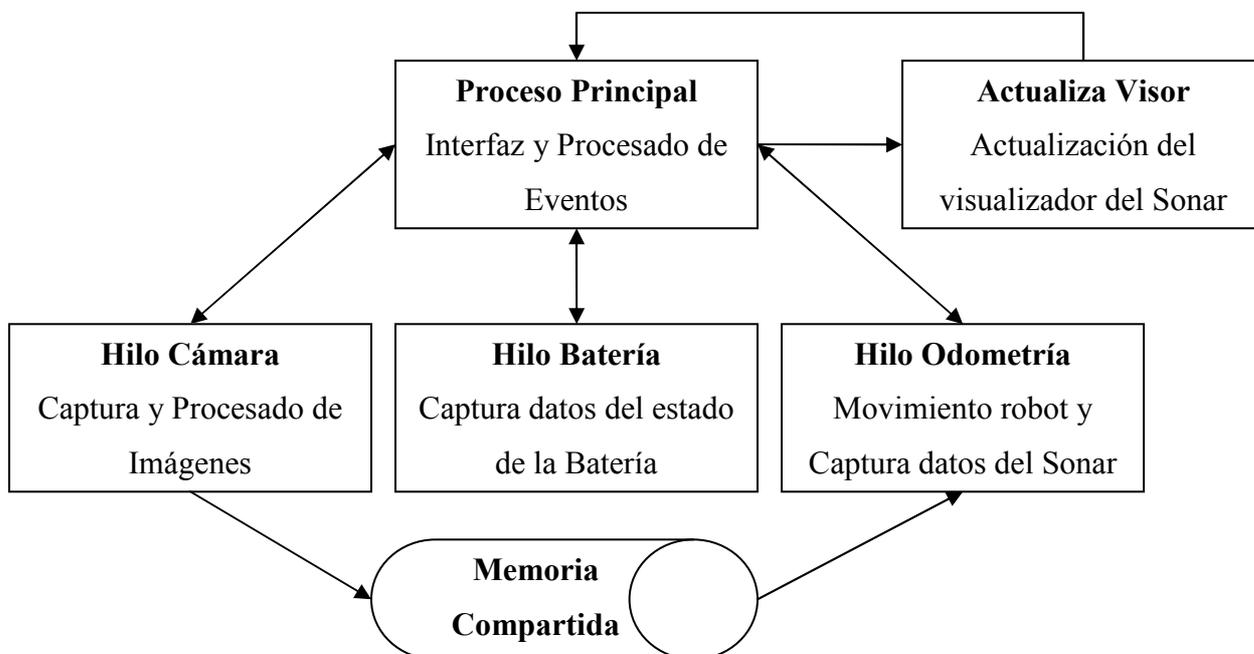


Figura 3 – Esquema general de la aplicación

En él se muestran los procesos en los que se dividió el proyecto.

Se concibió una estructura en la que se crea un hilo de procesamiento para cada toma de datos, debido a la falta de capacidad de cómputo de los sistemas monoprocesador actuales, los cuales no pueden realizar de forma paralela una gran cantidad de tareas tan intensivas como pueden ser la actualización de imágenes en pantalla simultáneamente con su captura y la actualización del resto de datos recogidos por los distintos sensores.

Se creó un Proceso Principal, encargado de realizar la mayor parte del trabajo visual de la aplicación. Es él quien crea el resto de hilos de procesamiento, los cuales toman los diferentes datos de los sensores del robot y muestra gráficamente esta información, actualizándola periódicamente. También se encarga de la gestión de los eventos enviados por cada uno de estos hilos de procesamiento, permitiendo la correcta sincronización de información entre ellos.

El proceso Hilo Cámara es el encargado de la obtención de las imágenes recogidas por la cámara montada en el robot y su procesado en busca de patrones haciendo uso de técnicas de Realidad Aumentada mediante la biblioteca ARToolkit. Cuando identifica un patrón en las imágenes, le envía la información relacionada al Hilo Odometría, encargado del envío de órdenes de movimiento al robot para su correcta navegación guiada a través de estos datos recibidos del *Hilo Cámara*. Este envío de información se realiza a través de Memoria Compartida, mecanismo de intercambio de información entre distintos procesos. Además de este cometido de movimiento del robot, *Hilo Odometría* también toma los datos procedentes de los sonares de los que dispone dicho robot para realizar una correcta navegación, evitando los obstáculos que encuentre durante su movimiento entre patrones, y enviándolos a Actualiza Visor, para su representación gráfica y posterior devolución al *Proceso Principal* para su visualización.

Para la recepción de datos relativos al estado de la batería del robot se creó el proceso Hilo Batería, que los envía al *Proceso Principal* para su visualización.

#### **4.1. Sistemas de Navegación del Robot Móvil**

Los dos sistemas de navegación desarrollados se fundamentan en un mismo principio de navegación híbrida, en el que el robot se mueve por el entorno sin colisionar con los obstáculos que detecta, mediante el uso de sus sensores, en busca de un patrón de entre los que tiene previamente cargados en memoria, comportamiento que conforma la parte reactiva del sistema.

La parte deliberativa entra en juego cuando uno de estos patrones es detectado en el flujo de imágenes tomadas por la cámara, en cuyo caso, mediante técnicas de Realidad Aumentada que extraen información acerca de la orientación y posición relativa entre la cámara y el patrón, el robot se dirige hacia él, evitando los obstáculos encontrados en el trayecto entre ambos. Es en este punto cuando se diferencian los comportamientos de ambos principios de navegación:

- Navegación cuasireactiva, llamada así porque muestra un comportamiento reactivo hasta recibir un estímulo visual (imagen), que gracias a su procesado mediante técnicas de Realidad Aumentada, dota al robot de la

capacidad de realización de tareas más complejas que el simple deambular sorteando obstáculos. En este tipo de navegación, una vez alcanzada la posición en la que se encuentra el patrón, el robot gira 90° para continuar la búsqueda de otro patrón, repitiendo el proceso.

- Navegación topológica asociativa. En este tipo de navegación, una vez alcanzada la posición en la que se encuentra situado el patrón, el robot ejecutará la acción que tiene asociado dicho patrón, principio en el que se fundamentan los métodos de navegación topológica asociativa: crear un comportamiento que convierte una observación sensorial (patrón determinado reconocido en las imágenes) en una dirección de movimiento determinada.

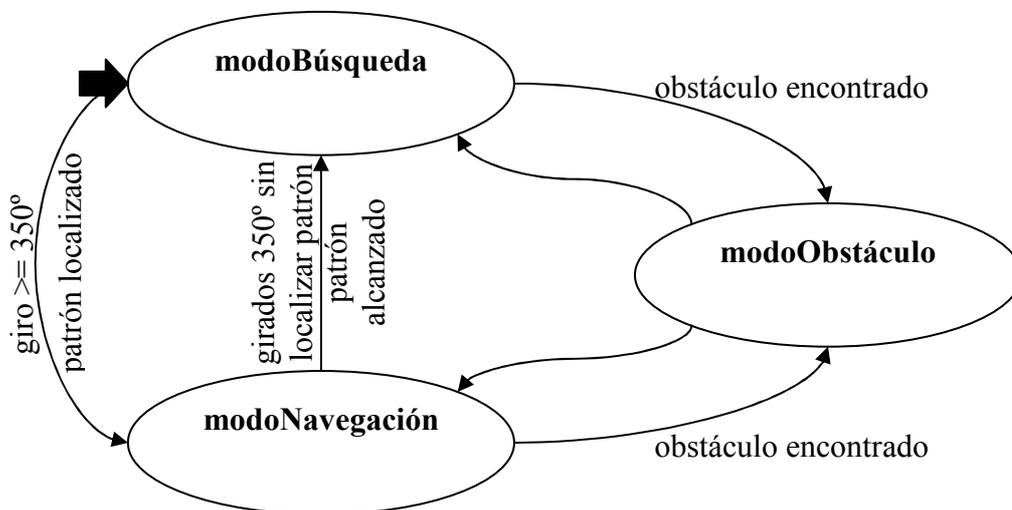


Figura 4 - Máquina de estados de los sistemas de navegación

En la máquina de estados mostrada en la Figura 4, se representa el funcionamiento en el que se basan ambos sistemas de navegación: en el modoBúsqueda el robot gira sobre sí mismo sin colisionar, ya que si encuentra un obstáculo pasa al estado modoObstáculo en el cual se procede a la evasión del mismo, hasta identificar un patrón en las imágenes tomadas por la cámara, con lo que pasaría al estado modoNavegación para alcanzarlo (también sin colisionar). Si gira más de 350° sin encontrar un patrón, pasa al estado *modoNavegación* para avanzar y volver al *modoBúsqueda* para repetir el proceso.

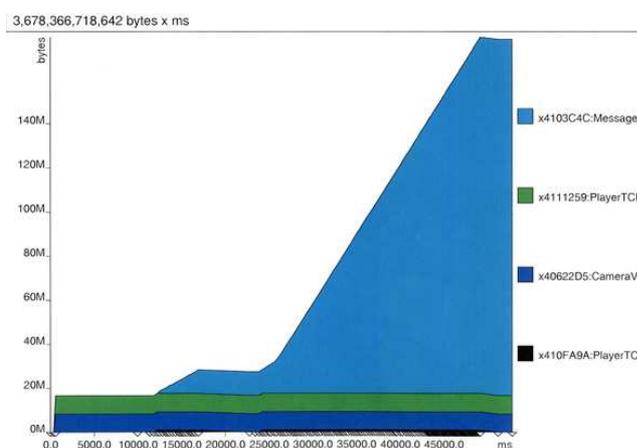
Para el reconocimiento de patrones en las imágenes tomadas por la cámara y la obtención de su posición respecto de ésta para su alcance por parte del robot, se hizo uso de técnicas de Realidad Aumentada proporcionadas por la biblioteca ARToolkit.

Esta biblioteca binariza las imágenes que recibe de una cámara, es decir, las convierte a una escala de grises, detecta los cuadrados que hay en ellas y los compara con los que tiene almacenados en memoria. Una vez reconocidos, proporciona su posición en un sistema de coordenadas centrado en la cámara mediante una matriz de transformación, lo que permite al robot alcanzar dicha posición.

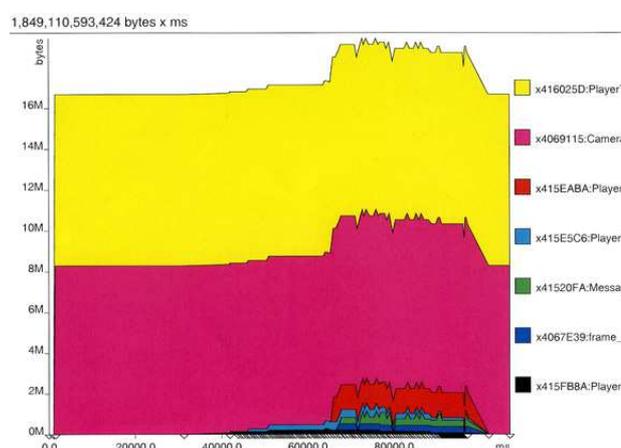
Tanto este movimiento del robot como la obtención de datos de cada uno de los sensores del robot se realiza gracias a la utilización del servidor multirobot Player, que proporciona un control simple y completo sobre los dispositivos sensores y actuadores que posee un robot a los programas cliente que se conecten a él.

Durante el desarrollo de este proyecto se detectó un error crítico en el núcleo del servidor Player. Este error consistía en un desbordamiento en la cola de mensajes del servidor, ya que no se eliminaban de memoria los mensajes sustituidos en la cola.

Este error pudo ser detectado gracias a la utilización de una aplicación de *profiling* y depuración para Linux, *Valgrind*. La aplicación *Massif* perteneciente a *Valgrind* es un analizador de memoria que determina el uso de memoria por parte de las distintas funciones/métodos del programa a analizar, y con el cual se detectó el método problemático.



**Figura 5 - Gráfico del uso de memoria del servidor Player antes de la aplicación del parche**



**Figura 6 - Gráfico del uso de memoria del servidor Player tras la aplicación del parche**

Como se puede observar en la Figura 5, el uso de memoria de la función en la que se creaban los mensajes aumentaba linealmente con el tiempo, lo que hacía inviable su utilización. Para su solución se tuvo que crear un parche a partir de archivos actualizados del repositorio de Player, en los que ya se había solucionado dicho fallo de

memoria. Tras la instalación del parche, se realizó una nueva prueba del servidor bajo la supervisión del analizador de memoria *Massif* y se obtuvo el gráfico adjunto en la Figura 6. Como se puede apreciar, el uso de memoria se mantiene estable en unos niveles adecuados y específicamente el uso de memoria de la función generadora de mensajes de datos descendió drásticamente, con lo que el problema fue resuelto satisfactoriamente.

Otro aspecto que se analizó fue la distancia a la que las marcas pueden ser detectadas por la biblioteca de Realidad Aumentada utilizada. Este aspecto determina la autonomía del robot.

En otro Proyecto de Fin de Carrera desarrollado en el Departamento de Informática y Automática de esta universidad se ha alcanzado una distancia de detección de 4 m con una cámara distinta a la utilizada en la realización del presente.

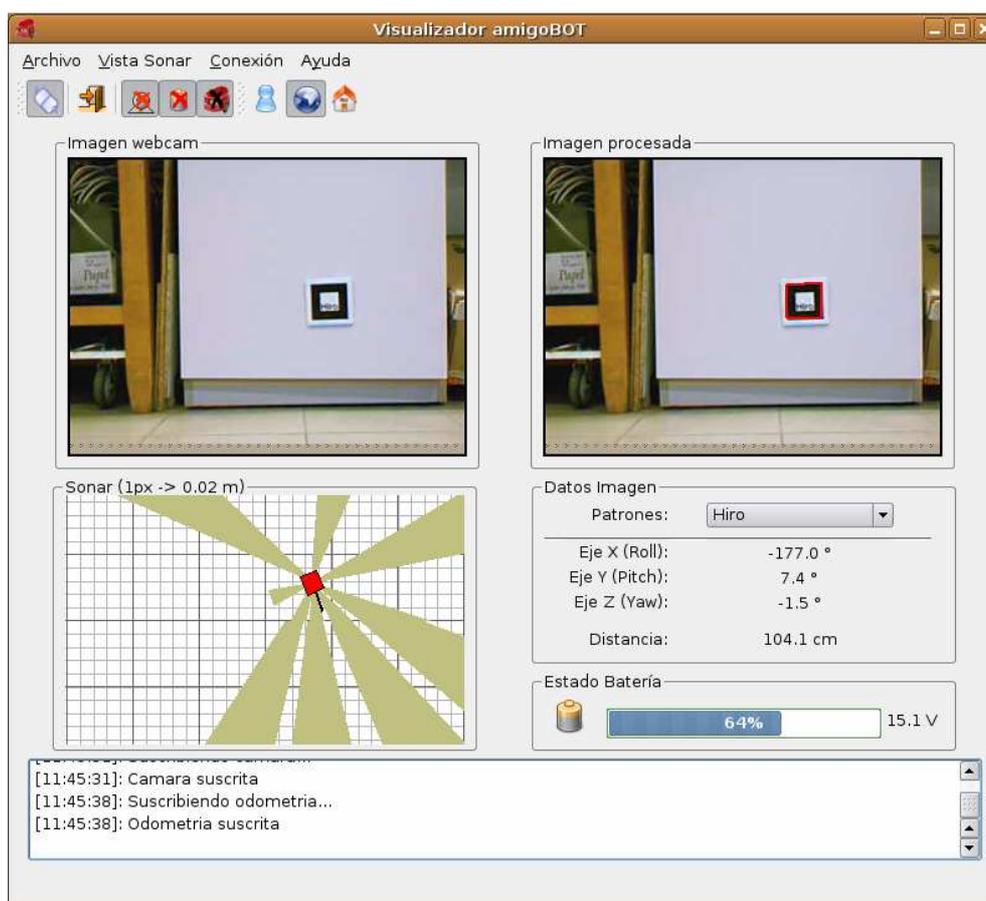


Figura 7 – Captura de pantalla de la interfaz gráfica de usuario desarrollada

## 5. Conclusiones

La conclusión a destacar en este punto es la implementación satisfactoria del sistema de navegación basado en el reconocimiento visual de patrones en tiempo real de un robot móvil autónomo, tanto en su modalidad *cuasireactiva* como topológica, objetivo principal del proyecto.

Se ha realizado el desarrollo completo que ha permitido validar y analizar las tecnologías utilizadas.