

Aplicación para un Gestor de Parquaderos Inteligentes.

Luis Felipe Vargas Rojas. 0836342.

Edgar Andres Moncada Taborda 0832294.

Heberth Vargas Tello. 1124798.

Presentado a:

Oscar Bedoya.

Universidad Del Valle.

Facultad De Ingeniería.

Ingeniería De Sistemas.

Cali

Noviembre 2011

Tabla de Contenidos

1. Requerimientos para la Aplicación	2
2. Sustentación de Cada Requerimiento	2
2.1. Archivo de Entrada y Actualización del Parqueadero	2
2.2. Selección del los Tipos de Búsqueda.	2
2.3. Sustentación para la Heurística Admisible	5
2.4. Lista de Movimientos en Interfaz	5
2.5. Información Referente al Algoritmo	5
3. Definición de la Meta	7

1. Requerimientos para la Aplicación

Esta aplicación se realizó tomando en cuenta los requerimientos establecidos en el enunciado del proyecto los cuales son:

1. Ingresar los datos de un ambiente determinado por medio de un archivo de texto que siga las convenciones dadas .
2. Desplegar gráficamente el parqueadero en su estado inicial, es decir, tal como se lee del archivo.
3. Seleccionar el tipo de algoritmo de búsqueda a aplicar: “No informada” o “Informada” .
4. Si se selecciona búsqueda “No informada” se puede elegir entre “Preferente por amplitud”, “Costo uniforme”, “Preferente por profundidad evitando ciclos”
5. Si se selecciona búsqueda “Informada” se puede elegir entre “Avara” y “A*”. Debe diseñar una heurística admisible.
6. Una vez aplicado un algoritmo se debe mostrar en la interfaz gráfica el conjunto de movimientos que realiza el gestor del parqueadero inteligente.
7. Después de aplicar un algoritmo se debe mostrar un reporte con la siguiente información: cantidad de nodos expandidos, profundidad del árbol y tiempo de cómputo

2. Sustentación de Cada Requerimiento

2.1. Archivo de Entrada y Actualización del Parqueadero

La aplicación a través de su interfaz nos permite ingresar un archivo de texto tipo txt (Requerimiento 1), esto se hace desde la barra de menú ubicada en la parte superior de la interfaz donde encontramos la etiqueta archivo, una vez ubicados ahí se da click y se despliega un menú en el cual encontramos 3 opciones entre las cuales esta cargar archivo al darle click en esa etiqueta nos despliega un frame que nos permite seleccionar el archivo una vez seleccionado el programa lo carga automáticamente en la interfaz (Requerimiento 2). ver figure 1,2,3.

2.2. Selección del los Tipos de Búsqueda.

Los requerimientos de selección del tipo de búsqueda y el algoritmo a usar en ella (Requerimiento 3 y 4) se satisfacen a través de pestañas colocadas en el menú superior a las cuales se acceden presionando click sobre la pestaña configuración paso siguiente nos despliega dos opciones búsqueda informada y no informada de la cual elegimos la que deseamos, al pasar el mouse sobre alguna de estas dos pestañas nos desplegara los algoritmos referentes a este tipo de búsqueda, y por ser radio button nos permite ver cual era el que estaba seleccionado, y si seleccionamos otro automáticamente quita el que se haya seleccionado anteriormente. figure 4.

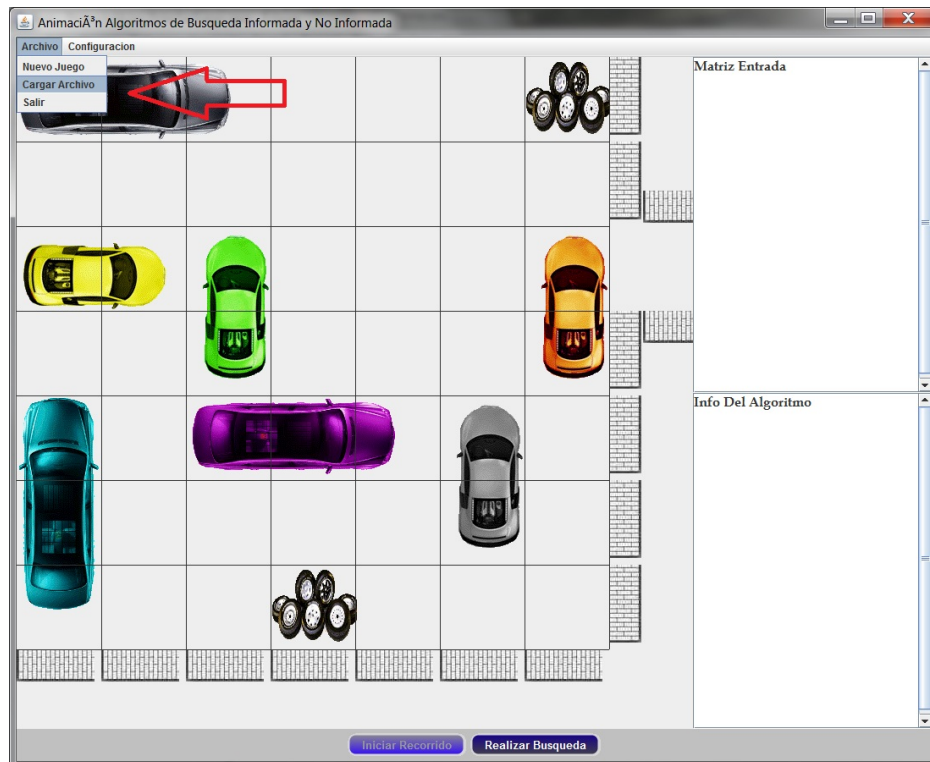


Figura 1: Explicación gráfica para desplegar el frame de selección de archivos de entrada.

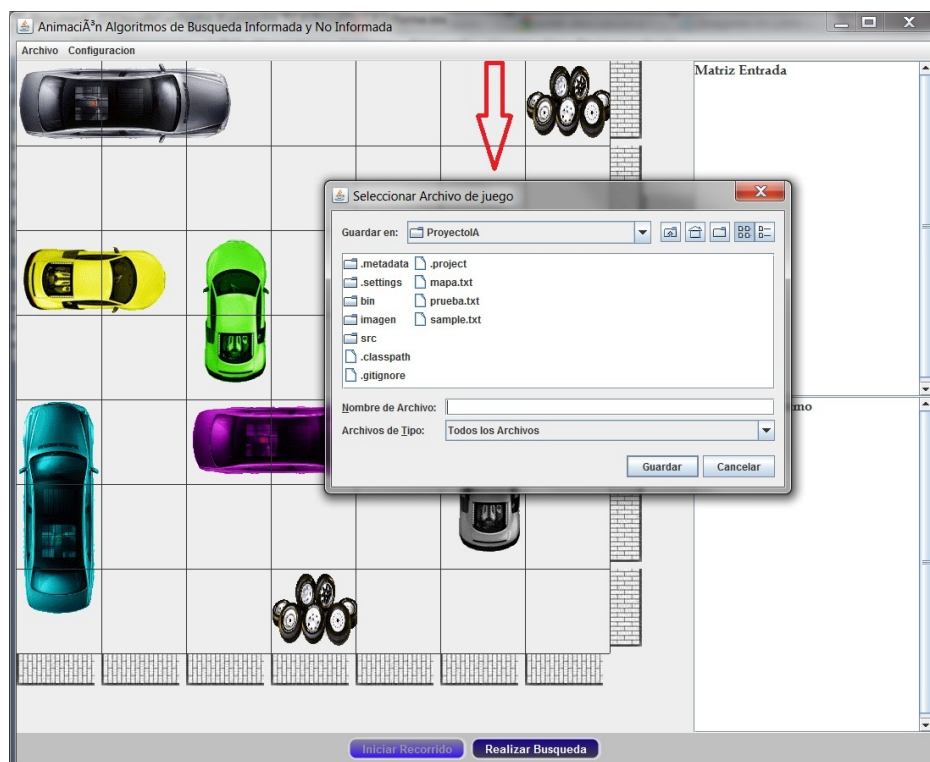


Figura 2: Explicación gráfica del frame de selección de archivo de entrada.

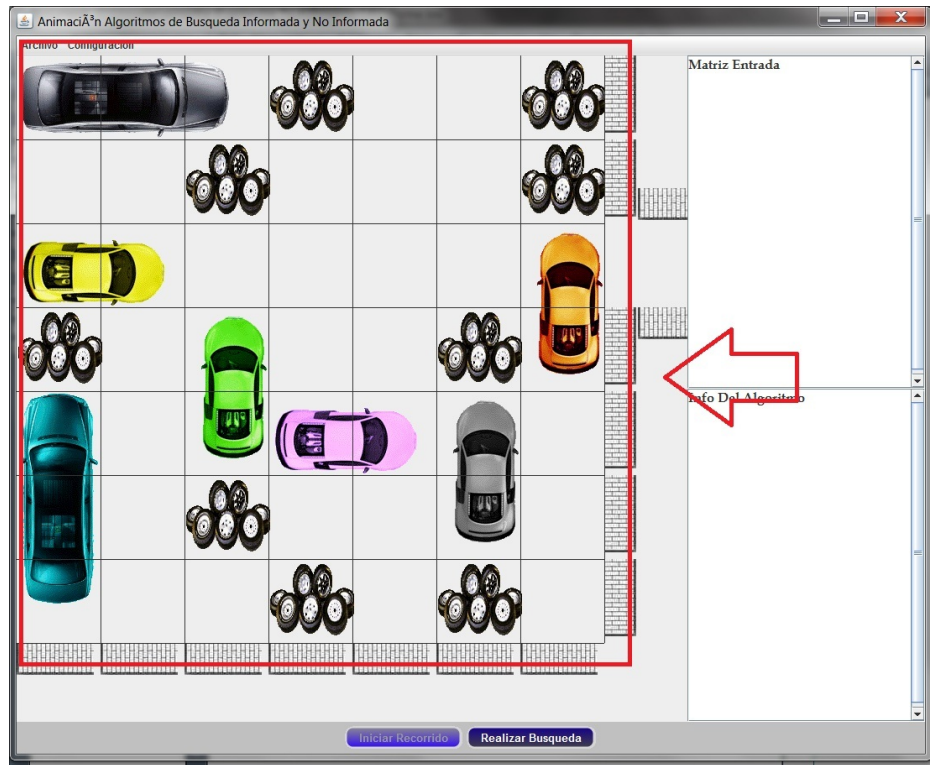


Figura 3: Explicación gráfica de la actualización del parqueadero al ingresar un nuevo archivo de entrada.

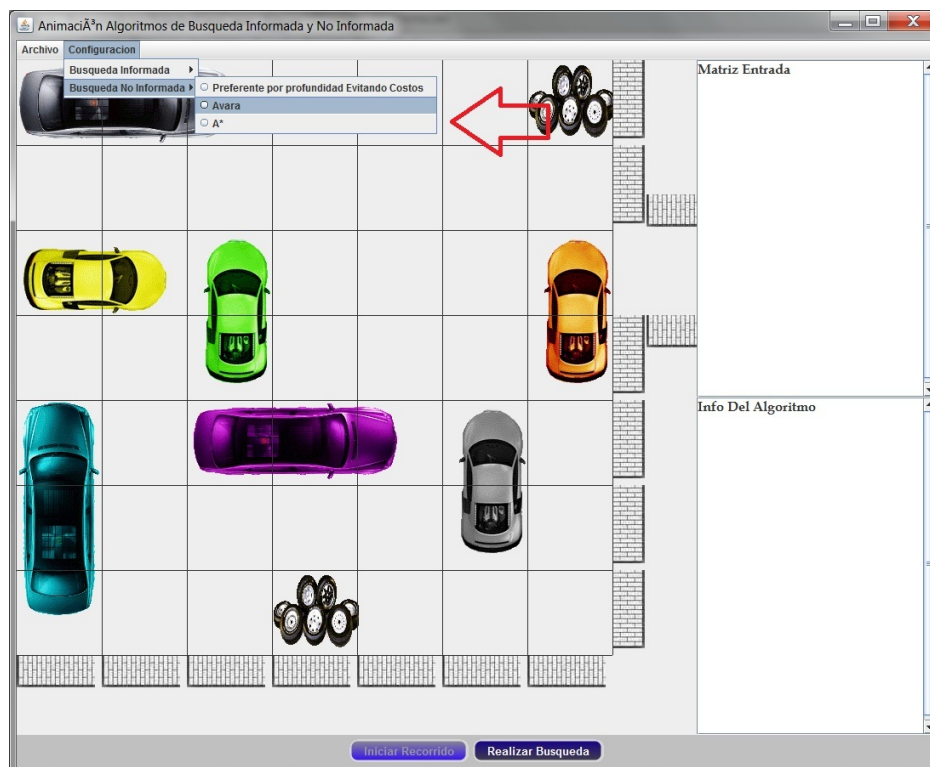


Figura 4: Explicación gráfica de las pestañas para la selección del algoritmo de búsqueda

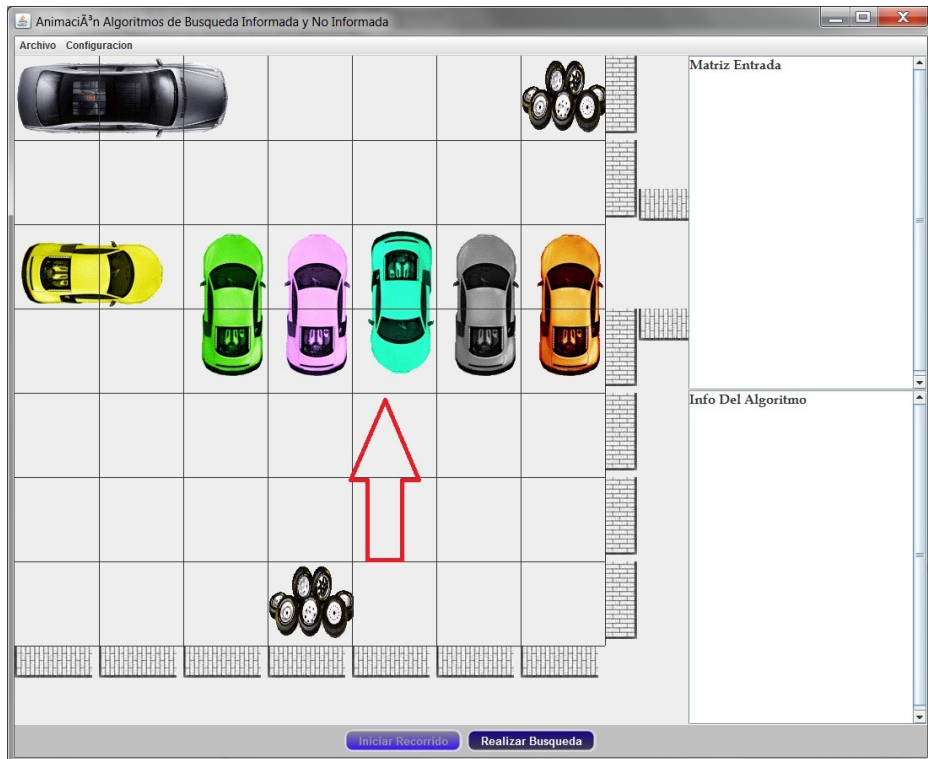


Figura 5: Explicación gráfica del peor caso de la heurística

2.3. Sustentación para la Heurística Admisible

El requerimiento 5 nos informa que se debe diseñar una heurística admisible, cuando se habla de una heurística admisible es por que la misma es menor o igual que el costo real de la solución.

Se decide que la heurística admisible es la cantidad de carros que obstaculizan la salida del carro solución en este caso "A".

Es admisible por que en el peor caso que es cuando todas las casillas las ocupe un carro el costo sera igual al costo de la solución real. (ver figure 5)

2.4. Lista de Movimientos en Interfaz

El requerimiento 6 el cual nos dice que se debe mostrar la lista de movimientos con los parámetros especificados en el enunciado se satisface a través de la interfaz de usuario y lo podemos ver en la figure 6.

2.5. Información Referente al Algoritmo

Finalmente el requerimiento 7 que nos pide que mostremos la cantidad de nodos expandidos, profundidad del árbol y tiempo de cómputo. Lo podemos observar en la figura 7.

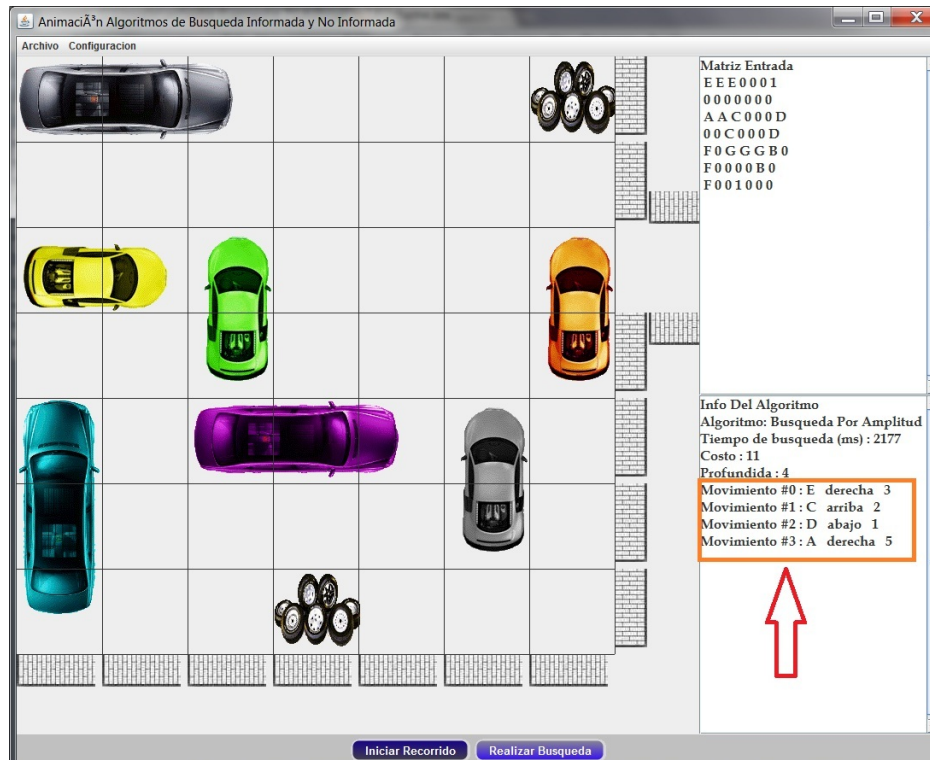


Figura 6: Lista de movimientos a realizar una vez aplicado el algoritmo de búsqueda

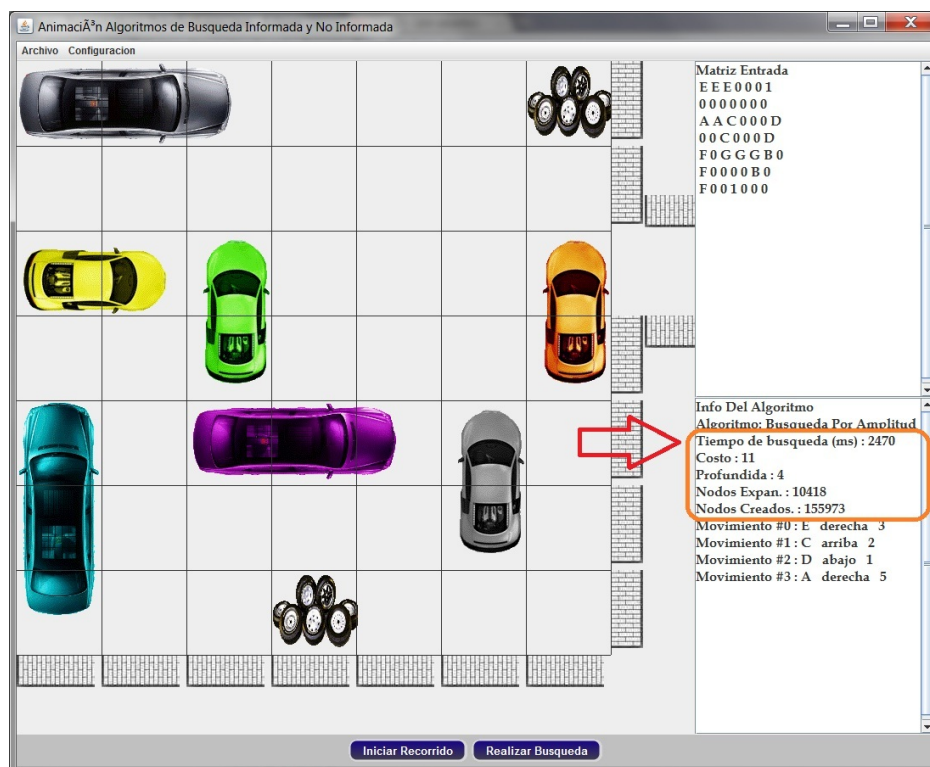


Figura 7: Información referente al algoritmo utilizado.

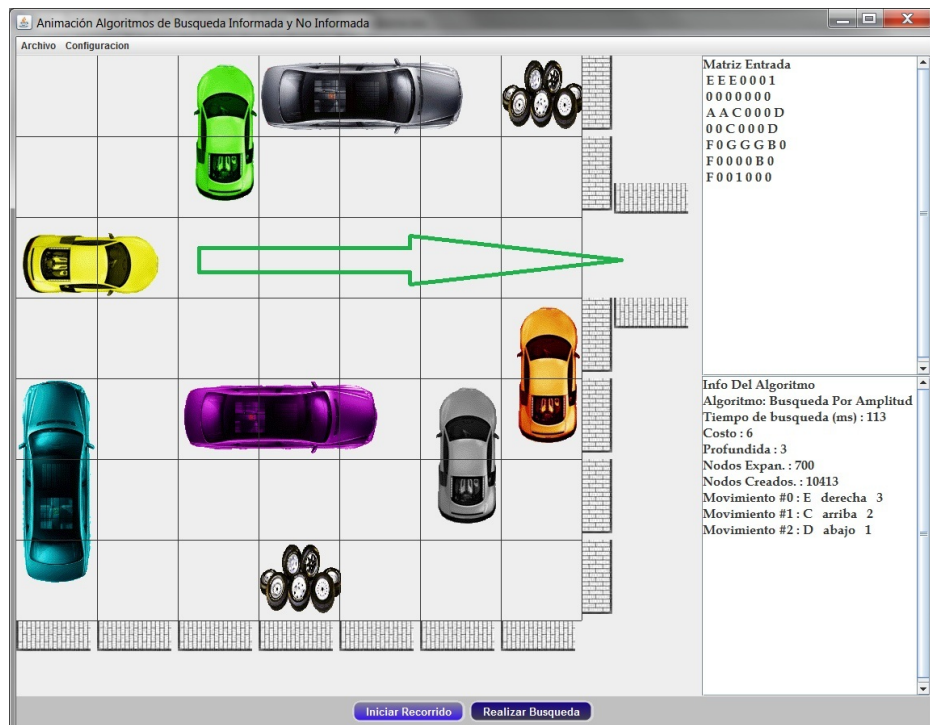


Figura 8: Caso para el cual en la implementación es meta.

3. Definición de la Meta

Cabe resaltar que para la definición de la meta y procurando que el algoritmo sea mas optimo se decide que la meta se logra cuando el carro A no tenga un obstáculo adelante para seguir. Esto se hace a nivel de implementación ya que gráficamente se muestra como si esto no pasara y la meta fuera cuando el carro sale esto nos ahorra una gran cantidad de nodos dependiendo del algoritmo. ver figure 8.