

MEMORIA PRÁCTICA 1 - INGENIERÍA DEL CONOCIMIENTO

DETALLES DE IMPLEMENTACIÓN

Se trata de una aplicación **Electron**, elegí esta tecnología ya que una forma sencilla podía desarrollar aplicaciones multiplataformas de manera sencilla. Otro factor que tuve en cuenta para desarrollar una aplicación multiplataforma es muchas de las aplicaciones más populares son aplicaciones **Electron**, esto me daba varias ventajas significativas, como una documentación de calidad y una amplia comunidad que responde a las preguntas frecuentes del desarrollo.

Una de las ventajas que da **Electron** es que ejecuta **NODEJS** que para mí es una ventaja significa ya que esto significa que se van a poder utilizar tecnologías web dentro de Electron, algo que en este momento prefiero ya que desarrollar con JAVA me cuesta.

Ya que usamos **NODEJS** y todo Electron está basado en tecnologías WEB, los lenguajes de programación que use han sido: **JAVASCRIPT, HTML, CCS**.

Para manejar los paquetes y dependencias en proyectos Node lo normal hasta ahora es usar **NPM**, pero en su lugar yo elegí **YARN** ya está destinado a convertirse en el nuevo estándar para manejar paquetes y dependencias, además que la velocidad de ejecución es menor en comparación con npm.

Durante los últimos años estamos viendo el auge de las tecnologías web, donde podemos encontrar muchos frameworks que nos ayudan a que nuestras aplicaciones se vean bien.

Elegí **React Native** como framework para la aplicación debido a lo fácil que es implementar componentes.

En resumen, esta es una aplicación **Electron**, que dentro corre un proceso **NodeJs** que nos permite ejecutar el framework **React Native** qué es el que nos muestra toda la parte visual y el que se encarga de ejecutar la lógica del modelo.

Procedimiento

El procedimiento que he seguido para implementar el Algoritmo A* ha sido entender cómo funciona el algoritmo gracias a la teoría dada en clase y las transparencias facilitadas.

Una vez entendido gracias a las distintas Estructuras de Datos que tiene **Javascript** he implementado el algoritmo con **Arrays, Maps y colas de prioridad**.

MANUAL DE USUARIO

El proceso de ejecución de la aplicación es muy sencillo, ejecutamos la aplicación (PracticaAStar.exe) en nuestro sistema operativo Windows, aceptamos ejecutar aplicaciones desconocidas (se muestra esto porque en la build no hay posibilidad de firmar, ya que está no es una aplicación oficial).

La pantalla que veremos al iniciar la aplicación es esta:



Para poder calcular un camino tenemos que indicar a la app los puntos que queremos calcular.

- Primero indicamos el punto de inicio, hacemos click en el botón **INICIO**, para indicar el inicio del camino, seleccionamos una casilla en el tablero y en este

se marca la letra **I**. Mientras el botón de inicio está seleccionado podemos cambiar la casilla de inicio.

- Para poner obstáculos en el tablero seleccionamos el botón **OBSTÁCULOS**, ponemos todos los que deseamos.
- Por últimos seleccionamos el botón **FIN** para indicar la casilla de destino en el camino.
- Podemos reiniciar el tablero con el botón **RESET**.
- Para poder encontrar el camino de nuestro tablero seleccionamos el botón **ENCONTRAR CAMINO**. La aplicación calcula el camino y lo muestra con O

Si falta por colocar algún elemento o hay algún problema en la pantalla aparece un mensaje de Error. Podemos reiniciar el tablero siempre que queramos con el botón **RESET**.