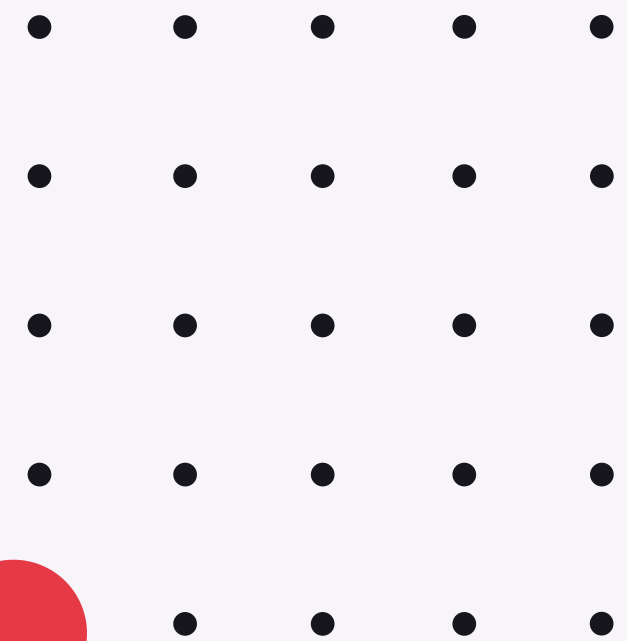
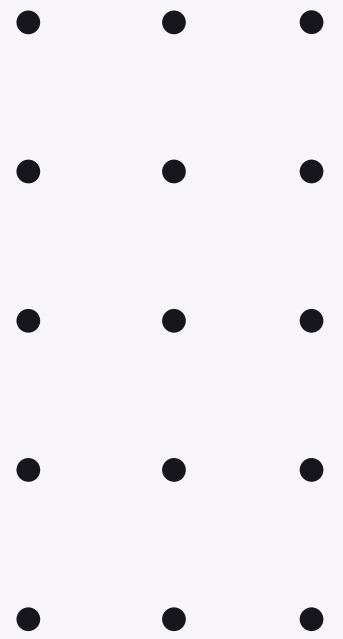


Cobertura del tablero de Ajedrez

Guarnieri Victoria - Velloso Manuela
Laboratorio de Programación II





Nuestro Algoritmo

Optimización del tablero

El tablero fue dividido en cuatro secciones, donde se colocaron las piezas de forma estratégica, de modo tal que estas cubren la mayor cantidad de casillas. Estas secciones fueron almacenadas en listas.

Posicionamiento

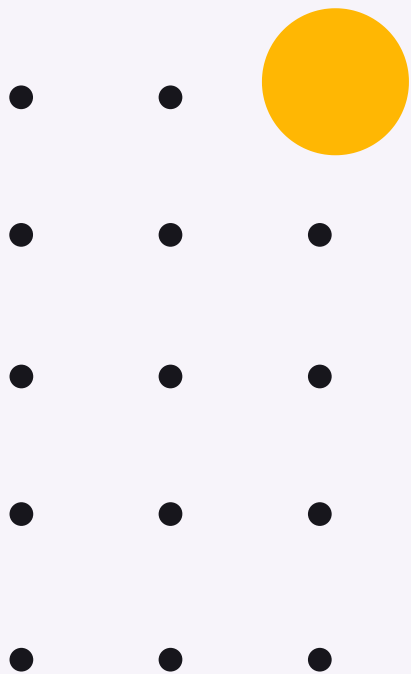
Se posicionan las fichas de acuerdo a las secciones, y se hacen las consideraciones especiales: todas surgen de su posición inicial en una partida de Ajedrez. Las fichas se almacenaron en una SortedList.

Ataque

Se usa un método polimórfico, y se marcan las casillas del tablero que están siendo atacadas tanto levemente como fatalmente.

Verificación

Se descartan los tableros que no son solución y, en caso contrario, se clasifican en leves o fatales.





Ventajas

● **SortedList**

Se utilizó una SortedList para guardar las fichas que se van posicionando en el tablero. Se sabe perfectamente dónde está cada ficha.

● **Color alfiles**

El código garantiza que los alfiles se posicionen en casillas con distinto color, por lo que toda solución parte de las posiciones iniciales de una partida de Ajedrez.

● **Superposición**

El algoritmo permite que un caballo y una torre se superpongan para resolver el problema.

● **Verificar distintas**

La función Contains, de librería, fue utilizada para determinar qué soluciones eran repetidas y así poder descartarlas.

● **Bordes**

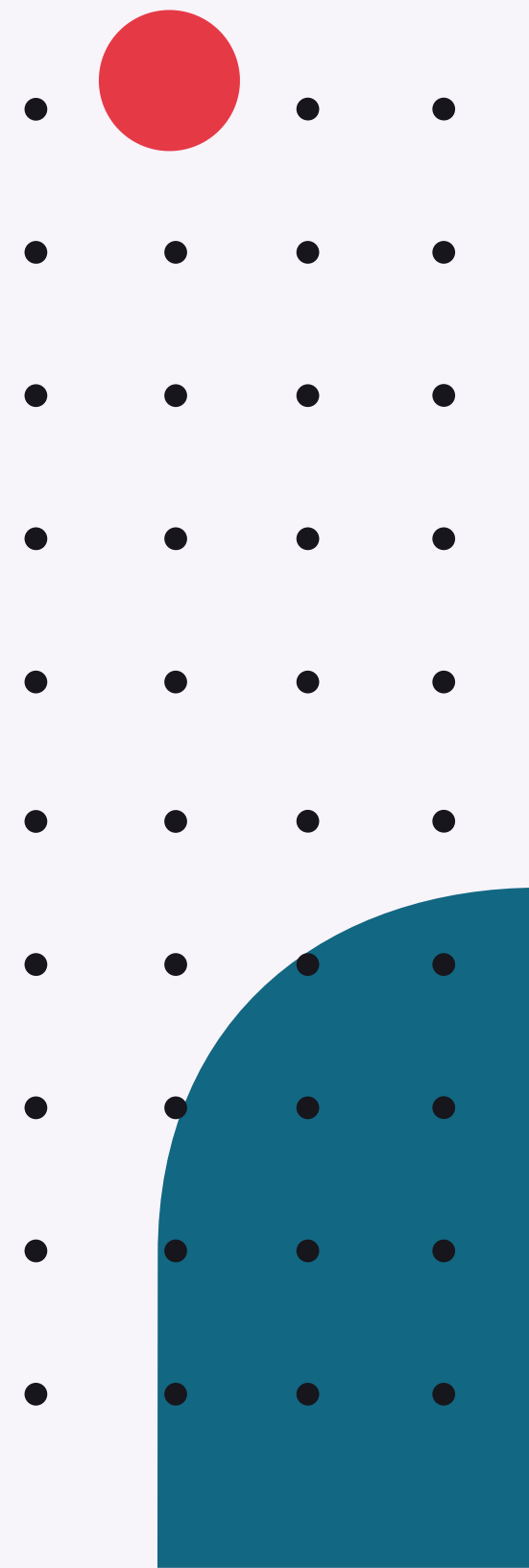
El algoritmo asegura que todas las fichas que se posicionan se encuentran dentro del tablero.

● **Complejidad**

La cota inferior de nuestro algoritmo es $\Omega(n) = n$.

● **DataGrid**

Utilizamos un DataGridView para imprimir los tableros.





iGracias!

Guarnieri Victoria - Velloso Manuela
Laboratorio de Programación II