



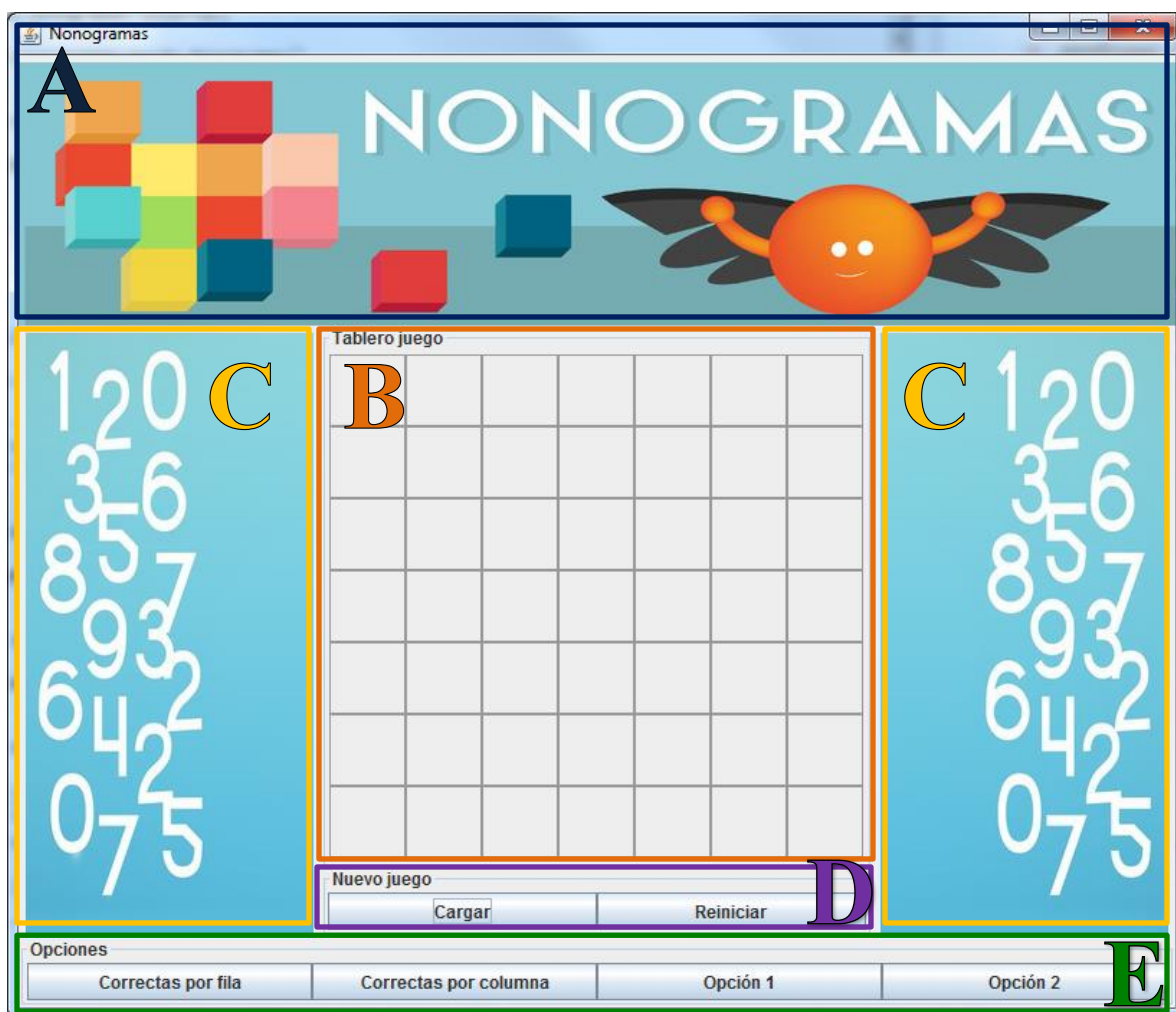
Ejercicio de nivel 6

Nonogramas

Consideraciones adicionales de diseño

Consejos prácticos para la construcción de la interfaz gráfica de la aplicación

1. Recomendación para la distribución de los paneles en la ventana principal.



Cada rectángulo señalado en la imagen anterior corresponde a un panel de la siguiente forma:

- A. Panel con la imagen del banner.
- B. Panel con la información del tablero: El tablero es una matriz de 7x7, donde las primeras dos filas y columnas corresponden a las pistas.
- C. Paneles auxiliares para las imágenes laterales.
- D. Panel con las opciones para un juego nuevo.
- E. Panel con las opciones adicionales.

2. Para incluir una imagen en un botón, se utiliza un `ImageIcon` de la siguiente manera:

```
JButton btnImagen = new JButton ( );  
ImageIcon icono= new ImageIcon(rutaImagen);  
btnImagen.setIcon( icono );
```

Donde **rutaimagen** es la ruta donde se encuentra ubicada la imagen. Por ejemplo, la imagen de la casilla rellena se encuentra en " data/imagenes/casilla_rellena.png".

Lectura de datos

A continuación, se describen unos métodos que pueden serle útiles:

1. El `Split` se utiliza para separar las cadenas delimitadas por un conjunto conocido de caracteres en sub-cadenas. Por ejemplo, si se tiene una cadena de caracteres que tiene valores separados por comas (","), se puede usar el método `split` para obtener cada uno de estos valores. Este método retorna un arreglo de cadena de caracteres, y en cada posición del arreglo está la sub-cadena que estaba separada por la coma en la cadena inicial.

Por ejemplo, se tiene al siguiente cadena

```
String cadenaADividir = "Hoy:quiero:estudiar";
```

Para obtener las subcadenas divididas por dos puntos (":"), se usa el siguiente comando.

```
String[] cadenaDividida = cadenaADividir.split(":");
```

El arreglo resultante tendrá tres posiciones, y cada uno tendrá los siguientes valores:

- `cadenaDividida[0]` : "Hoy";
- `cadenaDividida[1]` : "quiero";
- `cadenaDividida[2]` : "estudiar";

2. El método `charAt()` permite obtener un carácter localizado en una posición dada de una cadena de caracteres.

Por ejemplo, se tiene la siguiente cadena:

```
String cadena = "cupi2";
```



Si se desea obtener el carácter que se encuentra en la posición 2 de la cadena, se usa el siguiente comando:

```
char elemento= cadena.charAt(2);
```

El carácter asignado a la variable elemento será el carácter 'p'.

