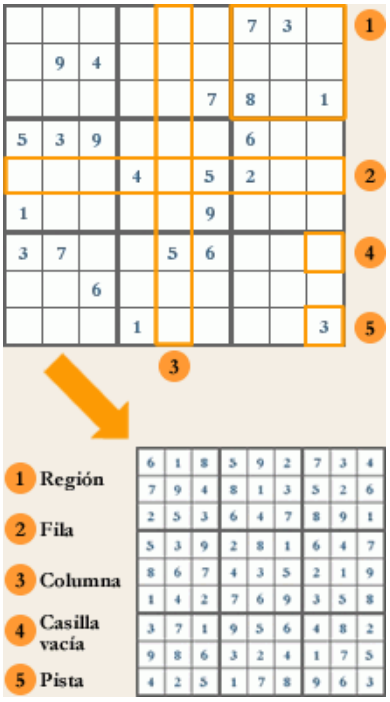
Manual PySudoku

Iván Aveiga
Kevin Campuzano

6 de agosto de 2013

1. Información

Sudoku es un juego en donde en una cuadrícula de 9x9 no debe de tener números repetidos en filas, columnas o subcuadrícula de 3x3.



[1]

2. Aspectos Técnicos

Implementamos un juego de Sudoku en que el usuario debe resolver un tablero de sudoku según la difucultad elegida, entre estas tenemos.

2.1. Herramientas de Desarrollo:

- **Lenguaje de Programación:** Python 2.7 64 bits.
- **Libreria Gráfica:** PyQt 4 64 bits.
- **Sistema Operativo:** Windows 7 64 bits.
- **Sistema de Versionamiento:** Github.
- **Herramienta de Documentación:** Doxygen.
- Fácil: 36-41 casillas llenas.

[1] http://www.sudokumania.com.ar/sm1/images3/reglas.png

- Normal: 32-35 casillas llenas.
- Avanzado: 28-31 casillas llenas.
- Experto: 23-27 casillas llenas.

2.2. Algoritmo

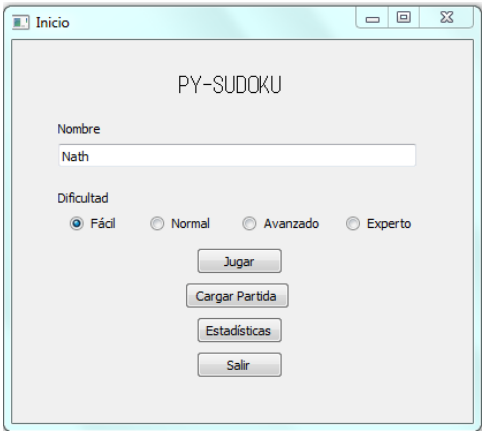
La forma de generar un tablero es usando un algoritmo que genere un tablero de 9x9 completo, una vez que se ha generado el tablero se procede a eliminar celdas aleatoriamente, el número depende de la dificultad. Para el juego se manejan dos tableros de sudoku, uno completo que se mantiene de fondo para la comparación y las ayudas, y el tablero en que el usuario jugará.

Generación de un tablero lleno

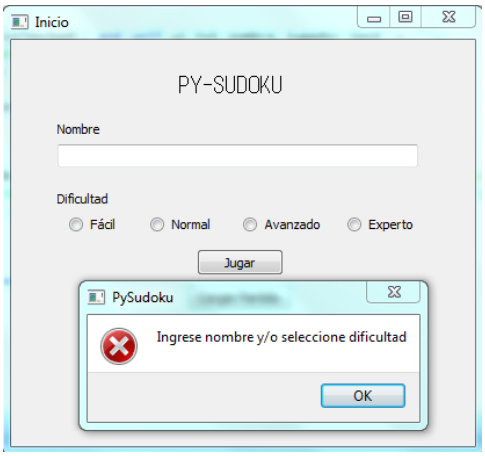
```
proc poblar_matriz( )
    matriz = [9][9]
    col = 0
    mientras  col < 9:
        posibles = [1,2,3,4,5,6,7,8,9]
        mezclar(posibles)
        fila = 0
        prueba = 0
        mientras fila < 9 y prueba < 200
            malas = 0
            parar = 0
            para cada elemento i en posibles
                si posibles[i] está en columna
                    malas += 1
            si malas == 9
                limpiar matriz
                parar = 1
                fila = 0
                prueba += 1
            si parar == 0
                numero = aleatorio (1,9)
                mientras conflicto(numero)
                    numero = aleatorio(1,9)
                matriz[fila][col] = numero
                remover numero de posibles
                filas + = 1
        col += 1
        si prueba == 20
            limpiar matriz
            col = 0
    retornar matriz
```

2.3. Interfaz Gráfica

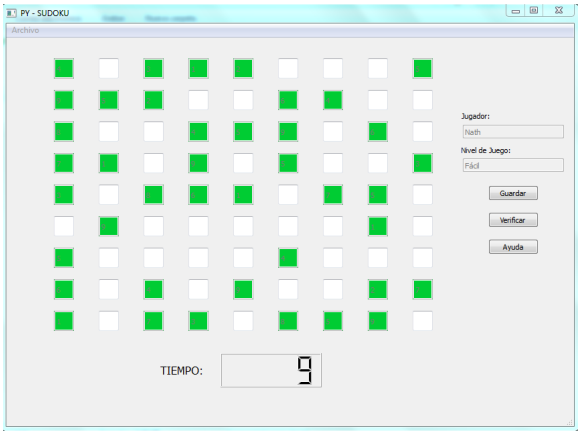
Al iniciar la aplicación se muestra una ventana donde ingresar el nombre y la dificultad a jugar.



Mientras no se ingrese los datos de nivel y el nombre no se podrá iniciar el juego, saltando un mensaje de error.



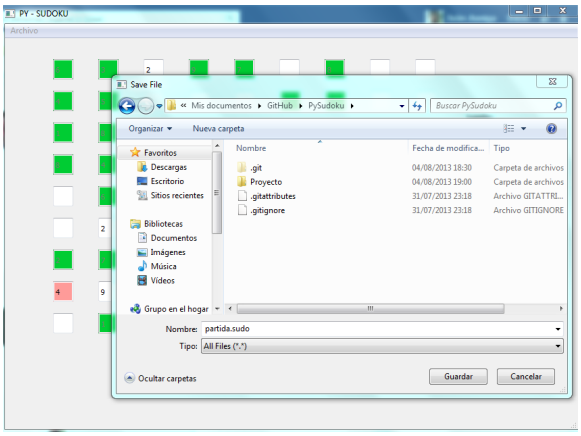
Una vez ingresado el nombre y seleccionada la dificultad aparece la ventana a jugar, mostrada a continuación.



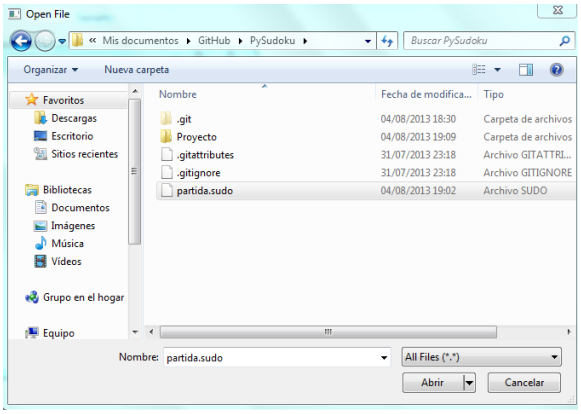
2.4. Funcionalidad

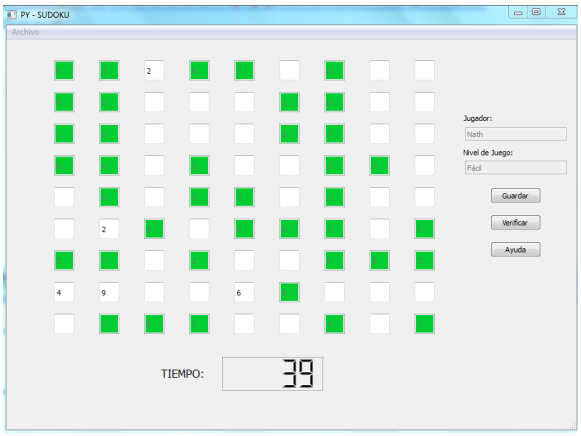
El jugador cuenta con un botón de ayuda, son 5 ayudas, una vez usada las ayudas se deshabilita este botón, si el jugador usa por lo menos una ayuda queda eliminada la opción de participar en el ranking de mejores jugadores. El juego genera una celda aleatoria y muestra el valor correcto, además se deshabilita la edición de dicha celda y su color se torna turquesa.

El jugador puede guardar la partida en cualquier momento y a su vez cargarla.

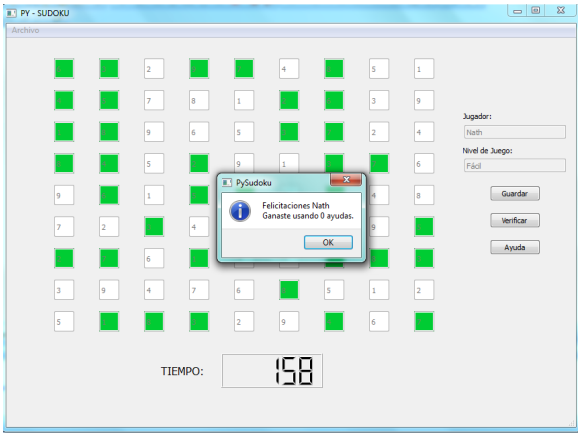


Se puede ver el proceso de carga de una partida guardada:

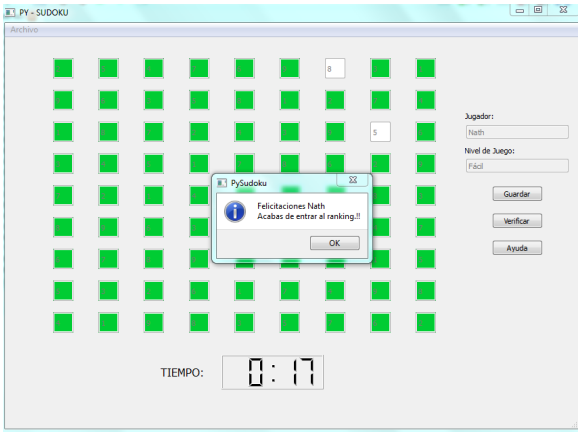




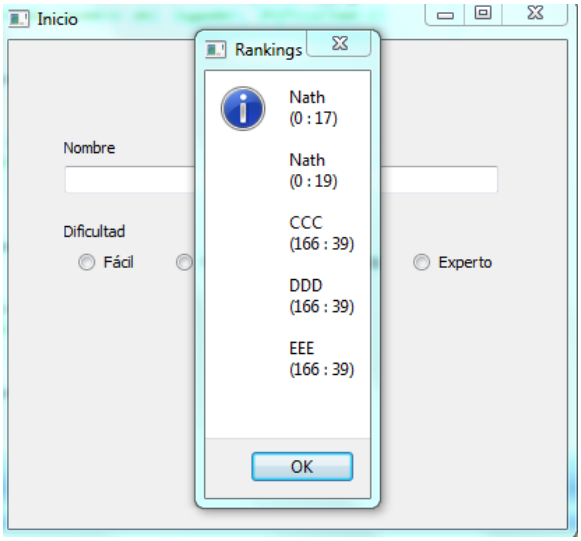
Al resolver correctamente el tablero el cronómetro se paraliza y se muestra el mensaje de que el usuario ganó.



El puntaje se maneja en base al número de segundos que toma resolver el Sudoku, si es un buen puntaje entra al ranking, en esta imagen podemos ver el mensaje de ingreso al ranking.



El usuario puede consultar el ranking, donde se muestra los mejores 5 tiempos.



3. Conclusiones

Python es un lenguaje que presta muchas facilidades, una de ellas es la facilidad de escribir código de manera casi natural, el tipado dinámico ayuada en muchas cosas, PyQt el binding de Qt para Python da la facilidad del diseño de interfaces con el Qt Designer. A Diferencia del proyecto pasado (implementación en C++) se logró cumplir las especificaciones del proyecto.

La especificación de guardar / cargar una partida cifrada se resolvió usando serialización y guardando el archivo en binario, esto garantiza que el usuario no pueda entender el contenido al abrir la partida con un editor de texto plano, además que para ver el contenido debe conocer la estructura del objeto serializado.