

SC++ TECH CHALLENGE

Descripción del reto

En este reto, los participantes deberán resolver un juego bien conocido por todos y con unas reglas muy particulares: un SUDOKU. Solucionarlo puede ser muy fácil, pero también muy difícil si no se eligen las técnicas de programación adecuadas y las estructuras de datos correctas.

En Scalian queremos encontrar una manera rápida de resolver este juego tan popular.

¿Eres capaz de resolver un sudoku a través de una herramienta programada por ti?

El ganador tendrá una recompensa que le permitirá seguir creciendo profesionalmente y, ¡quién sabe, quizá podría ser en Scalian!

Contenido

1.- ¿Quiénes somos y por qué estamos aquí?.....	2
2.- El reto	2
3.- Cuáles son los requisitos y cómo lo haremos	2
4.- Cronograma y evaluación de soluciones.....	3
5.- Recompensas.....	4
6.- Memoria Técnica.....	5
6.1.- Reglas del juego	7
6.2 Interacción con el tablero	8
6.3 Resolver Sudoku	9
6.4 Borrar Sudoku	9
6.5 Resultado.....	9
6.6.- Interacción con la UI desde el código	10
7.- Mejoras propuestas.....	10
7.1.- Resultado vitaminado	10
 Imagen 1.- Kit de iniciación a la electrónica.....	 5
Imagen 2.- Ficheros de proyecto de Qt Creator.....	6
Imagen 3.- SC++ Tech Challenge - Pantalla principal	7
Imagen 4.- Configuración de una casilla del tablero	8

1.- ¿Quiénes somos y por qué estamos aquí?

[Scalian Spain](#) es una multinacional tecnológica de origen francés que cuenta con más de 550 empleados en España y más de 20 años de experiencia en el sector. Esta compañía es especialista en estrategia e innovación, con capacidades metodológicas y digitales, así como en soporte operativo, todo ello orientado a mejorar el rendimiento de los negocios.

Más concretamente, Scalian se enfoca en proporcionar **soluciones disruptivas del mundo del dato e inteligencia artificial**, que permiten a las empresas aprovechar las últimas tecnologías para que puedan ser competitivas en aspectos como eficiencia, sostenibilidad, innovación, diversidad y reputación. Con oficinas en Málaga, Madrid Alicante y Barcelona, Scalian acompaña en **el proceso de transformación digital a clientes de sectores** como banca y servicios financieros, salud, industria, transporte, ferroviario y automoción, con **soluciones innovadoras** que cubren el ciclo end to end de sus productos.

Scalian Spain ahora se une al Campus 42 de Málaga para retar a sus alumnos a resolver a través de programación uno de los juegos más populares del mundo: el **SUDOKU**.

2.- El reto

El origen del Sudoku, que se significa «número único» en japonés, se remonta a la década del 1980 cuando el editor Maki Kaji rediseño este puzle a partir del rompecabezas Numbre Place de Howard Garns. De esta manera, nació el juego de Sudoku tal cual lo conocemos a día de hoy y se calcula que actualmente más de 100 millones de personas juegan con regularidad ([1](#)).

La digitalización llega a muchas partes de nuestra vida cotidiana, y los juegos populares no son excepción. De jugar a sudokus en revistas y periódicos, ahora jugamos en aplicaciones. Pero, y si lo llevas aún más lejos, **¿serás capaz de resolver un sudoku a través de una herramienta programada por ti?**

El objetivo de este reto es encontrar el algoritmo que resuelva este popular juego en el menor tiempo posible y de la manera más eficiente posible. ¿Os atrevéis con este reto?

3.- Cuáles son los requisitos y cómo lo haremos

Para desarrollar la solución al reto que os proponemos usaremos **el lenguaje de programación C++ y el Qt** ([2](#)), un framework de propósito general que posiblemente no conozcáis.

Como acabamos de mencionar, Qt es un framework de propósito general y que es ampliamente usado en el desarrollo de aplicaciones que utilizan interfaz gráfica de usuario (GUI). Con Qt, también se pueden desarrollar herramientas que no necesitan una GUI (software para servidores, por ejemplo), aunque este no será nuestro objetivo.

C++ es bien conocido por todos vosotros ya que es uno de los lenguajes de programación de referencia en las formaciones impartidas en Campus42.

Sabiendo esto, el día que comience el reto impartiremos un pequeño seminario sobre el desarrollo de aplicaciones con Qt y pondremos a vuestra disposición una pequeña interfaz gráfica que hemos diseñado y que vais a tener que aprender a usar. Sentiros libres de modificar esta interfaz gráfica como deseéis, para adaptarla a vuestro gusto, pero siempre deben aparecer los elementos/controles que componen la interfaz original de partida.

Además del seminario, en el apartado [6.- Memoria Técnica](#) os proporcionamos una memoria técnica suficiente para entender el código que os proporcionamos, así podréis comenzar a resolver el reto que os proponemos lo antes posible y con el menor número de dudas.

4.- Cronograma y evaluación de soluciones

Puedes consultar la planificación completa del evento accediendo a nuestro [calendario interactivo](#).

La fecha límite de entrega del reto será el **01 de diciembre** a las **11:45PM**. A cada participante se le proporcionará un usuario y contraseña para el Moodle de Scalian Spain, en el cual podrán subir el reto antes de la fecha límite.

Por otra parte, la evaluación de las soluciones se llevará a cabo atendiendo a la siguiente rúbrica (ver Tabla 1):

Tabla 1.- Rúbrica

	MAL	ACEPTABLE	BIEN	EXCELENTE
FUNCIONAMIENTO	El programa no funciona	El programa básico funciona, pero presenta 3 o más defectos.	El programa básico funciona, pero presenta menos de 3 defectos.	El programa funciona sin defectos.
ESTRUCTURAS DE DATOS	Utiliza estructuras de datos poco eficientes	Utiliza estructuras de datos aceptables, pero	Utiliza las estructuras de datos adecuadas, pero no las	Utiliza las estructuras de datos adecuadas y las

	y/o no adecuadas	no son las más adecuadas.	utiliza de manera correcta.	usa de manera correcta.
ALGORITMO	El algoritmo utilizado no es el adecuado para este tipo de problemas.	El algoritmo utilizado es correcto, pero no es el mejor.	El algoritmo utilizado es correcto, pero presenta errores de concepto y/o no es óptimo.	El algoritmo utilizado es correcto, no presenta errores de concepto y es óptimo.
ENTREGA	Entrega la solución 10 minutos antes de la fecha límite.	Entrega la solución y se ubica en el cuartil Q3.	Entrega la solución y se ubica en el cuartil Q2.	Entrega la solución y se ubica en el cuartil Q1.
MEJORAS	No implementa ninguna de las mejoras propuestas.		Implementa alguna de las mejoras propuestas.	Implementa todas las mejoras propuestas.

5.- Recompensas

De entre los participantes, se seleccionará **un ganador** siguiendo la rúbrica del Apartado 4, Tabla 1.

El ganador recibirá de premio un [Miuzei Starter Kit](#) (valorado en 40€, ver Imagen 1). Este kit de iniciación es la puerta de entrada a un emocionante mundo de proyectos basados en Arduino. Con componentes como una placa R3, tutorías en español, breadboard, servomotor, suministro de energía de 9V/1A, sensores, Leds y mucho más; tendrás todo lo que necesitas para comenzar tus propios proyectos. El **lenguaje de programación está basado en C/C++**, por lo que podrás adquirir **conocimientos básicos de programación, electrotecnia y electrónica** ([3](#)).



Imagen 1.- Kit de iniciación a la electrónica

¡Pero espera, hay más! El ganador no sólo recibirá un fantástico kit de merchandising de Scalian, sino que también tendrá la oportunidad de ser considerado en futuros procesos de selección de Scalian Spain. ¡Tu talento podría abrirte puertas hacia un emocionante futuro profesional!

Prepárate para el desafío, demuestra tus habilidades y llévate a casa este increíble premio.

6.- Memoria Técnica

El proyecto que Scalian entrega a los estudiantes para resolver el reto consta de una serie de ficheros propios y otros de Qt (ver Imagen 2).

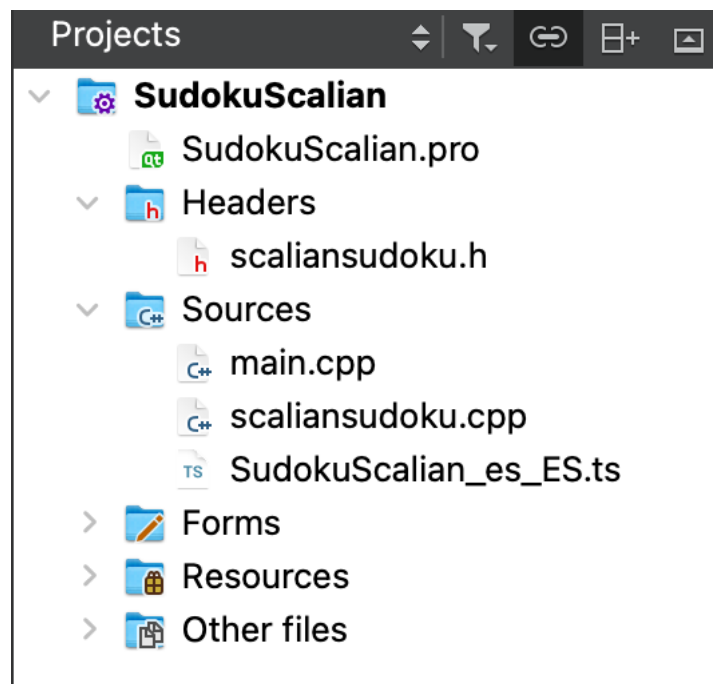


Imagen 2.- Ficheros de proyecto de Qt Creator

De los ficheros disponibles, sólo será necesario completar, a priori, el fichero ***scaliandsudoku.cpp*** (y su correspondiente cabecera) que se encuentra en el directorio *Sources*. Este fichero fuente contiene las definiciones de los métodos miembro de la clase ***ScalianSudoku*** cuyo fichero de cabecera (declaraciones) es ***scaliandsudoku.h*** y que se encuentra en el directorio *Headers*.

Teniendo esto en cuenta, si ejecutamos el proyecto utilizando Qt Creator aparecerá la siguiente pantalla (ver Imagen 3):

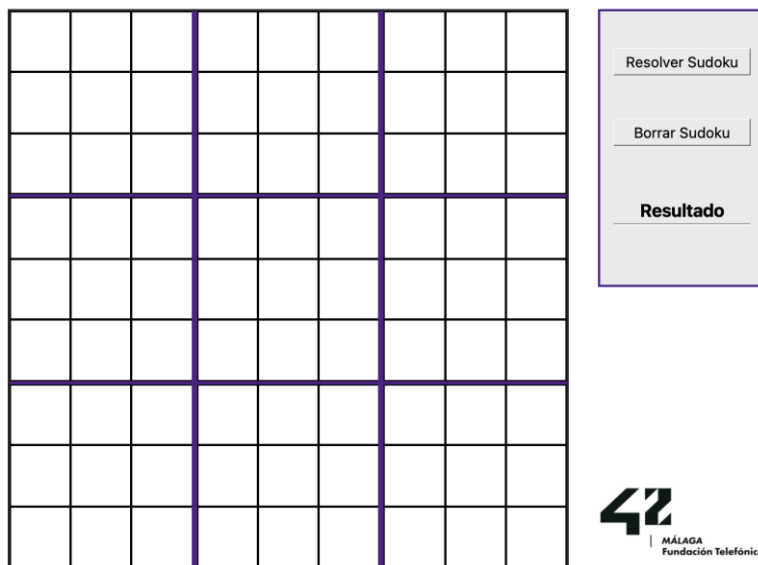


Imagen 3.- SC++ Tech Challenge - Pantalla principal

Como se puede apreciar (ver Imagen 3), la pantalla principal se divide en dos zonas bien diferenciadas:

1. Una cuadrícula (tablero) de tamaño 9x9 y que representa el tablero básico (sólo se pueden usar números del 1 al 9 para resolver el juego) de un Sudoku.
2. Una zona de control que presenta 2 botones (ver ítems a continuación) y una etiqueta para informar al usuario con el resultado (correcto o incorrecto) obtenido al resolver el tablero:
 - a. *Resolver Sudoku*
 - b. *Borrar Sudoku*

6.1.- Reglas del juego

El juego del Sudoku parte de **una cuadrícula de 9x9** casillas, que a su vez se divide en **9 regiones de 3x3** casillas. Partiendo de **algunos números ya dispuestos** en algunas de las casillas (números fijos determinados por el usuario), hay que **completar las casillas vacías con dígitos del 1 al 9 sin que se repitan por fila, columna o región**.

Reglas:

- Hay que completar las casillas vacías con un solo número del 1 al 9.
- En una misma fila no puede haber números repetidos.
- En una misma columna no puede haber números repetidos.

- En una misma región no puede haber números repetidos.
- La solución de un sudoku es única.

6.2 Interacción con el tablero

El juego del Sudoku parte de un tablero con algunas casillas cuyos valores son fijos y establecidos por el usuario antes de comenzar la partida. Para fijar un valor en una casilla basta con hacer doble clic sobre la casilla a la que se desea asignar el valor. Después de esta acción, la pantalla principal toma la siguiente forma (ver Imagen 4):

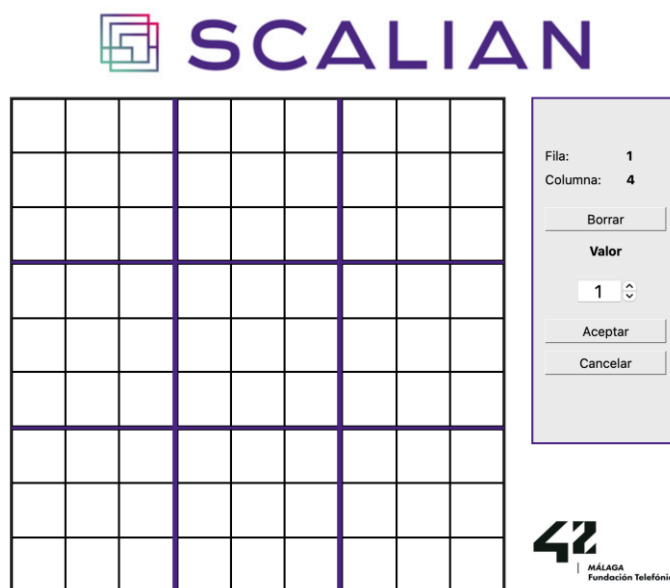


Imagen 4.- Configuración de una casilla del tablero

Como se puede apreciar en la Imagen 4, la zona de control cambia y muestra lo siguiente:

- La fila (eje y) y la columna (eje x) que se está editando.
- Un botón *Borrar* que permite borrar un valor previamente fijado por el usuario.
- Un control que permite seleccionar el valor que se va a asignar a la casilla.
- Un botón *Aceptar* para confirmar la asignación.
- Un botón *Cancelar* para salir del modo edición.

Teniendo esto en cuenta, debe saber que:

1. Al pulsar el botón *Borrar* se ejecutará la siguiente función (TO DO):

```
void borrarCelda(uint filaId, uint colId);
```

2. Al pulsar el botón *Aceptar* se ejecutará la siguiente función (TO DO):

```
void setearCelda(uint filaId, uint colId, uint valor);
```

NOTA: es responsabilidad del alumnos definir las estructuras de datos necesarias para resolver el reto.

6.3 Resolver Sudoku

Una vez que el usuario a establecido los valores de partida en las celdas elegidas será necesario pulsar sobre el botón *Resolver Sudoku* en la pantalla principal (ver Imagen 3). Al clicar sobre este botón se ejecutarán las siguiente funciones (TO DO):

```
void resolverSudoku();
```

```
bool chequearSudoku();
```

6.4 Borrar Sudoku

Para borrar el Sudoku el usuario debe clicar en el botón Borrar Sudoku que aparece en la pantalla principal (ver Imagen 3). Al clicar sobre este botón se ejecutará la siguiente función (TO DO):

```
void limpiarSudoku();
```

6.5 Resultado

El valor en la etiqueta *Resultado* que aparece en la pantalla principal (ver Imagen 3) se determina automáticamente en función del valor devuelto como resultado de llamar a la siguiente función:

```
bool chequearSudoku();
```

La siguiente tabla (ver Tabla 2) muestra el valor en la etiqueta en función del valor devuelto:

Tabla 2.- Valores en la etiqueta Resultado

VALOR DEVUELTO	TEXTO EN LA ETIQUETA	COLOR
true	Correcto	Verde
false	Incorrecto	Rojo

6.6.- Interacción con la UI desde el código

Para implementar correctamente el reto que se propone, el alumno necesitará interactuar con la interfaz gráfica proporcionada desde el código. Para esto, se proporciona la siguiente interfaz (se explicará durante el lanzamiento del reto):

```
bool limpiarCelda(uint filaId, uint colId);
bool escribirCelda(uint valor, uint filaId, uint colId, QColor color = QColor(Qt::GlobalColor::black));
void escribirResultado(const std::string &resultado, QColor color = QColor(Qt::GlobalColor::black));
```

7.- Mejoras propuestas

7.1.- Resultado vitaminado

La funcionalidad básica de la aplicación sólo contempla dos posibles valores como resultado de la comprobación del tablero (ver sección 6.5 Resultado). Teniendo esto en cuenta se propone que la etiqueta *Resultado* muestre los siguientes mensajes (ver Tabla 3):

Tabla 3.- Valores vitaminados en la etiqueta Resultado

TEXTO EN LA ETIQUETA	COLOR
Correcto	Verde
Incorrecto	Naranja/Amarillo
Imposible	Rojo