

# Reporte Sprint #0

## 1. Decisiones claves para el proyecto SOS

Lenguaje de programación orientado a objetos	Java
Librería GUI	Swing
IDE	IntelliJ IDEA 2023.1 (Community Edition)
Framework xUnit	Junit 5
Guía de estilo de programación	<a href="https://google.github.io/styleguide/javaguide.html">https://google.github.io/styleguide/javaguide.html</a>
Sitio de alojamiento del proyecto	<a href="https://github.com/WalterRGUni/CC-3S2/Proyecto-3S2">https://github.com/WalterRGUni/CC-3S2/Proyecto-3S2</a>

## 2. Pruebas unitarias

Programa:

```
TemperatureConverter.java x
1  package production;
2
3  3 usages
4  public class TemperatureConverter {
5
6      1 usage
7      public double CelsiusToFahrenheit(double celsiusTemperature) {
8          return (9.0 / 5.0) * celsiusTemperature + 32;
9      }
10
11      1 usage
12      public double FahrenheitToCelsius(double fahrenheitTemperature) {
13          return (5.0 / 9.0) * (fahrenheitTemperature - 32);
14      }
15  }
```

Pruebas unitarias:

```

TemperatureConverterTest.java x
1  package test;
2
3  import static org.junit.jupiter.api.Assertions.assertEquals;
4  import org.junit.jupiter.api.Test;
5  import production.TemperatureConverter;
6
7  public class TemperatureConverterTest {
8      @Test
9      public void CelsiusToFahrenheitTest() {
10         var converter = new TemperatureConverter();
11         double fahrenheit = converter.CelsiusToFahrenheit( celsiusTemperature: 100.0);
12         double expected = 212.0;
13         assertEquals(expected, fahrenheit, delta: 0.01);
14     }
15
16     @Test
17     public void FahrenheitToCelsiusTest() {
18         var converter = new TemperatureConverter();
19         double celsius = converter.FahrenheitToCelsius( fahrenheitTemperature: 100.0);
20         double expected = 37.78;
21         assertEquals(expected, celsius, delta: 0.01);
22     }
23
24 }

```

Resultados de las pruebas:

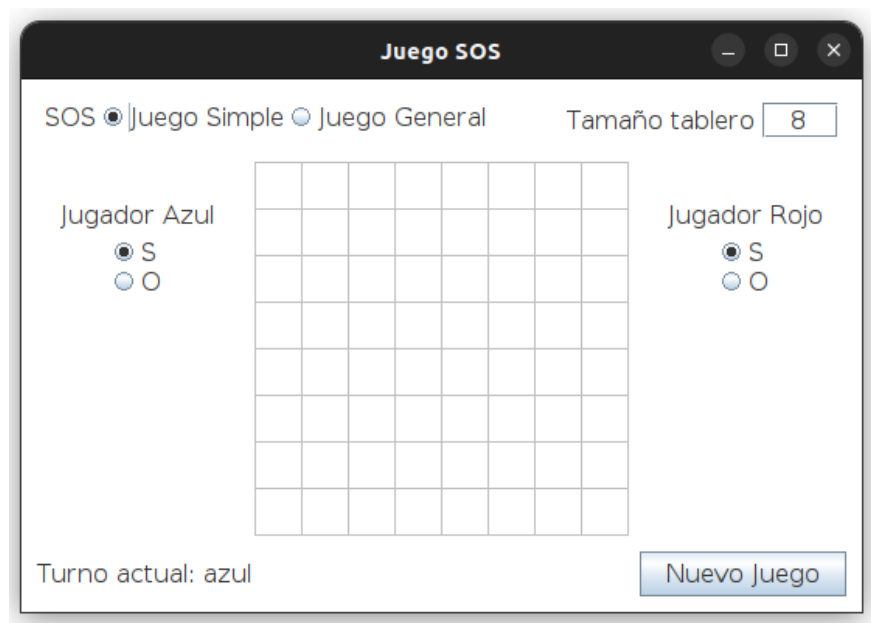
```

Run: TemperatureConverterTest x
[Icons] Tests passed: 2 of 2 tests – 50 ms
  ✓ TemperatureConverterTest (test) 50 ms /lib/jvm/java-17-amazon-corretto/bin/java ...
    ✓ CelsiusToFahrenheitTest() 48 ms
    ✓ FahrenheitToCelsiusTest() 2 ms
Process finished with exit code 0

```

### 3. Programación GUI

Captura de pantalla de la ejecución del programa:



Código fuente:

```
package produccion;

import java.awt.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;

public class SosGui extends JFrame {
    private Font fuente = new Font("SansSerif", Font.PLAIN, 16);
    private int tamañoTablero = 8; // número de celdas por lado en la cuadrícula
    private static final int TAMANIO_CELDA = 30;

    public SosGui() {
        setPanelDeContenido();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setTitle("Juego SOS");
        setVisible(true);
    }

    public void setPanelDeContenido() {
        JPanel panelCentral = new JPanel();
        panelCentral
```

```

        .setPreferredSize(new Dimension(tamanoTablero * TAMANIO_CELDA + 1,
            tamanoTablero * TAMANIO_CELDA + 1));

PanelSuperior panelSuperior = new PanelSuperior();

PanelInferior panelInferior = new PanelInferior();

PanelIzquierdo panelIzquierdo = new PanelIzquierdo();
panelIzquierdo.setPreferredSize(new Dimension(150, tamanoTablero * TAMANIO_CELDA));

PanelDerecho panelDerecho = new PanelDerecho();
panelDerecho.setPreferredSize(new Dimension(150, tamanoTablero * TAMANIO_CELDA));

Container panelDeContenido = getContentPane();
panelDeContenido.setLayout(new BorderLayout());

panelDeContenido.add(panelCentral, BorderLayout.CENTER);
panelDeContenido.add(panelSuperior, BorderLayout.NORTH);
panelDeContenido.add(panelIzquierdo, BorderLayout.WEST);
panelDeContenido.add(panelDerecho, BorderLayout.EAST);
panelDeContenido.add(panelInferior, BorderLayout.SOUTH);
}

private class PanelCentral extends JPanel {

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        setBackground(Color.WHITE);
        dibujarLineas(g);
    }

    private void dibujarLineas(Graphics g) {

        for (int fila = 0; fila <= tamanoTablero; fila++) {
            g.drawLine(0, fila * TAMANIO_CELDA, tamanoTablero * TAMANIO_CELDA, fila *

```

```

TAMANIO_CELDA;
    }

    for (int col = 0; col <= tamañoTablero; col++) {
        g.drawLine(col * TAMANIO_CELDA, 0, col * TAMANIO_CELDA, tamañoTablero * TAMANIO_CELDA);
    }
}
}

private class PanelSuperior extends JPanel {

    PanelSuperior() {
        setLayout(new BorderLayout());
        setBorder(new EmptyBorder(10, 10, 10, 10));
        setBackground(Color.WHITE);

        JPanel pnlTipoJuego = new JPanel();
        pnlTipoJuego.setBackground(Color.WHITE);

        // Elegir entre juego simple y general
        JLabel lblSos = new JLabel("SOS");
        lblSos.setFont(fuente);
        pnlTipoJuego.add(lblSos);

        JRadioButton btnJuegoSimple = new JRadioButton("Juego Simple", true);
        btnJuegoSimple.setBorder(BorderFactory.createEmptyBorder());
        btnJuegoSimple.setFont(fuente);
        btnJuegoSimple.setBackground(Color.WHITE);

        JRadioButton btnJuegoGeneral = new JRadioButton("Juego General", false);
        btnJuegoGeneral.setBorder(BorderFactory.createEmptyBorder());
        btnJuegoGeneral.setFont(fuente);
        btnJuegoGeneral.setBackground(Color.WHITE);

        pnlTipoJuego.add(btnJuegoSimple);
        pnlTipoJuego.add(btnJuegoGeneral);
    }
}

```

```

ButtonGroup btnGrpTipoJuego = new ButtonGroup();
btnGrpTipoJuego.add(btnJuegoSimple);
btnGrpTipoJuego.add(btnJuegoGeneral);

add(pnlTipoJuego, BorderLayout.WEST);

// Elegir tamaño de tablero
JPanel pnlTamañoTablero = new JPanel();
pnlTamañoTablero.setBackground(Color.WHITE);
JLabel lblTamañoTablero = new JLabel("Tamaño tablero");
lblTamañoTablero.setFont(fuente);
pnlTamañoTablero.add(lblTamañoTablero);
JTextField txtTamañoTablero = new JTextField(3);
txtTamañoTablero.setFont(fuente);
txtTamañoTablero.setText(String.valueOf(tamañoTablero));
txtTamañoTablero.setHorizontalAlignment(JTextField.CENTER);
pnlTamañoTablero.add(txtTamañoTablero);
add(pnlTamañoTablero, BorderLayout.EAST);
}

}

private class PanelInferior extends JPanel {
    PanelInferior() {
        setLayout(new BorderLayout());
        setBorder(new EmptyBorder(10, 10, 10, 10));
        setBackground(Color.WHITE);

        JLabel lblTurno = new JLabel("Turno actual: azul");
        lblTurno.setFont(fuente);
        add(lblTurno, BorderLayout.CENTER);

        JButton btnNuevoJuego = new JButton("Nuevo Juego");
        btnNuevoJuego.setFont(fuente);
        add(btnNuevoJuego, BorderLayout.EAST);
    }
}

```

```

private class PanelIzquierdo extends JPanel {
    PanelIzquierdo() {
        setBorder(new EmptyBorder(20, 20, 20, 20));
        setBackground(Color.WHITE);
        JLabel lblJugAzul = new JLabel("Jugador Azul");
        lblJugAzul.setFont(fuente);
        add(lblJugAzul);

        JRadioButton btnSAzul = new JRadioButton("S", true);
        btnSAzul.setFont(fuente);
        btnSAzul.setBackground(Color.WHITE);
        btnSAzul.setBorder(BorderFactory.createEmptyBorder());

        JRadioButton btn0Azul = new JRadioButton("0", false);
        btn0Azul.setFont(fuente);
        btn0Azul.setBackground(Color.WHITE);
        btn0Azul.setBorder(BorderFactory.createEmptyBorder());

        ButtonGroup btnGrpLetraAzul = new ButtonGroup();
        btnGrpLetraAzul.add(btnSAzul);
        btnGrpLetraAzul.add(btn0Azul);

        Box boxLetraAzul = Box.createVerticalBox();
        boxLetraAzul.add(btnSAzul);
        boxLetraAzul.add(btn0Azul);
        add(boxLetraAzul);
    }
}

```

```

private class PanelDerecho extends JPanel {
    PanelDerecho() {
        setBorder(new EmptyBorder(20, 20, 20, 20));
        setBackground(Color.WHITE);
        JLabel lblJugRojo = new JLabel("Jugador Rojo");
        lblJugRojo.setFont(fuente);
    }
}

```

```

add(lblJugRojo);

JRadioButton btnSRojo = new JRadioButton("S", true);
btnSRojo.setFont(fuente);
btnSRojo.setBackground(Color.WHITE);
btnSRojo.setBorder(BorderFactory.createEmptyBorder());

JRadioButton btnORojo = new JRadioButton("O", false);
btnORojo.setFont(fuente);
btnORojo.setBackground(Color.WHITE);
btnORojo.setBorder(BorderFactory.createEmptyBorder());

ButtonGroup btnGrpLetraRojo = new ButtonGroup();
btnGrpLetraRojo.add(btnSRojo);
btnGrpLetraRojo.add(btnORojo);

Box boxLetraRojo = Box.createVerticalBox();
boxLetraRojo.add(btnSRojo);
boxLetraRojo.add(btnORojo);
add(boxLetraRojo);
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new SosGui());
}
}

```