

Trabalho de Laboratório – Curso EI

Objetivos:

Introdução ao JavaFX.

Programas:

Pretende-se desenvolver o protótipo da interface gráfica de um smartwatch com capacidade para funcionar como: relógio digital (horas-minutos-segundos), ou como telefone com marcação por teclado.

Regras de implementação:

- Criar a aplicação utilizando o IDE NetBeans ou IntelliJ.
- Implementar o código necessário e testar no fim de cada nível.
- Use as convenções de codificação adotadas para a linguagem Java (ver **Notas**).

Implementação:

Nível 1:

- Crie uma aplicação JavaFX denominada **SmartClockApplication** que abre uma janela com 270x270 pixels.
- Descarregue do moodle as classes que suportam o funcionamento do relógio e guarde-as num package denominado **smartWatch**.
- Experimente o funcionamento do smartwatch em modo de consola.

Nível 2:

- Crie uma classe **SmartWatchPane** que herda de **BorderPane** e apresenta um design semelhante ao da **figura 1a** em que se mostra um botão “set mode” do lado direito do painel.
 - Para usar a fonte **DS-DIGIB** colocada no botão: crie uma pasta “resources” no seu projeto e dentro desta uma outra denominada “fonts”, descompacte o ficheiro com as fontes para essa pasta, em seguida defina um atributo do tipo **Font** e atribua-lhe uma **Font** definida com a instrução: `Font.loadFont("file:resources/fonts/DS-DIGI.ttf", 20);`
 - Para pintar de negro o background do botão crie um **Background** com a instrução: `Background blackBackground = new Background(new BackgroundFill(Color.BLACK, CornerRadii.EMPTY, Insets.EMPTY));` e em seguida atribua-o à respetiva propriedade do botão.
- Apresente o **SmartWatchPane** na **SmartClockApplication**.

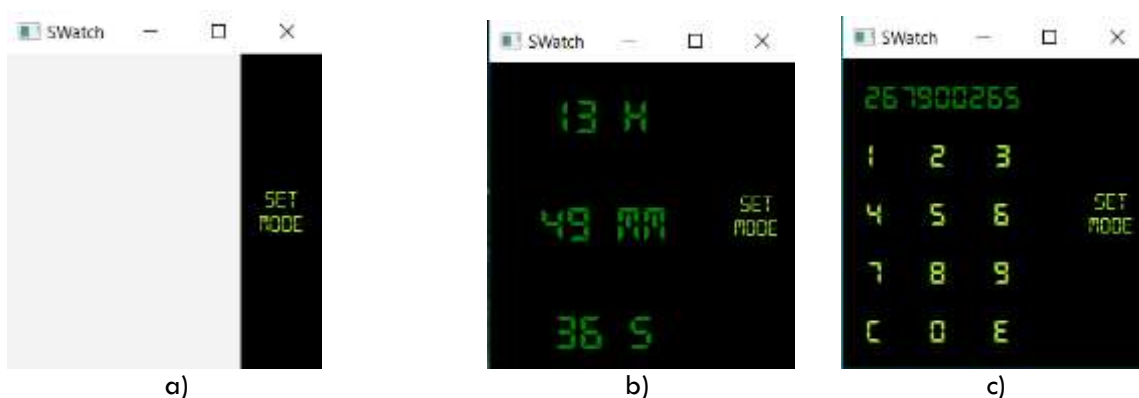


Figura 1: Imagens dos três painéis do relógio. (a) **SmartWatchPane**, (b) **ClockPane** no centro do **SmartWatchPane**, (c) **PhonePane** no centro do **SmartWatchPane**.

Nível 3:

- Crie uma classe **ClockPane** que irá preencher o centro do **SmartWatchPane** quando o relógio estiver em modo **Mode.CLOCK** e com um aspeto semelhante ao da **figura 1b**.
- O botão **Set Mode** irá permitir alternar o que é mostrado no centro, ou seja, estando em **Mode.CLOCK** deve estar a ver-se o relógio, quando se carrega no botão muda para modo **Mode.PHONE** e passa a ver-se o mostrador do telefone. Defina neste nível o comportamento do botão **Set Mode** para apresentar o **ClockPane** ao centro quando em modo **Mode.CLOCK** (Figura 1b) e nada quando em **Mode.PHONE** (Figura 1a).
- Apresente no ecrã ambas as opções.

Nível 4:

- Crie uma classe **PhonePane** com o aspeto da parte esquerda da **Figura 1c**.
- Defina o comportamento dos botões por forma a ser possível digitar um numero de telefone, usar a tecla C para apagar o mostrador e a tecla E para fazer a ligação.
- Experimente o relógio e o telefone.

Nível 5:

- Como se terá apercebido o relógio ainda não está a funcionar devidamente uma vez que não atualiza as horas automaticamente. Adicione um método à classe **ClockPane** que atualize o relógio a cada segundo. Pode usar um timer como é explicado em: <http://tomasmikula.github.io/blog/2014/06/04/timers-in-javafx-and-reactfx.html>
- Experimente o relógio e o telefone.

Notas:

Para os identificadores siga as convenções adotadas normalmente, em particular:

- 1) A notação **camelCase** para o nome das variáveis locais e identificadores de atributos e métodos.
- 2) A notação **PascalCase** para os nomes das classes.
- 3) Não utilize o símbolo '_', nem abreviaturas nos identificadores.