

Watch Face

Google I/O Extended em Juiz de Fora

Rafael Alves Feliciano

fael.anjelus@gmail.com

@rafaelmeteoro

github.com/rafaelmeteoro

- Graduado em Ciência da Computação pela UFJF
- Desenvolvedor Android na Appta
- 2 Anos de experiência em android
- Apoiador no GDG



Watch Face



Motivação

A circular smartwatch face with a pink upper half and white lower half. The pink section displays the time '05:20:57' and the date 'THURSDAY 19 FEB 2015'. The white section displays four icons: a speech bubble with '02 Messages', an eye with '05 km', a sun with '79.5 F Feels 80F', and a wind icon with '23 km/h'.

Material Face For Girls

Frillroid Watch Faces - 8 de abril de 2015 - 10

Personalização

Comprar: R\$3,36

+ Adicionar à Lista d

i Este app é compatível com alguns de seus dispositivos

Informações adicionais

Atualizado

8 de abril de 2015

Tamanho

12M

Instalações

50 - 100

Motivação



Clocki - Wear Watch Faces

azy - 10 de abril de 2015 - 10

Personalização

Comprar: R\$6,24



Adicionar à Lista de desejos

Este app é compatível com alguns de seus dispositivos.

★★★★☆ (422)

g+

Informações adicionais

Atualizado

10 de abril de 2015

Tamanho

5,6M

Instalações


10.000 - 50.000

Projeto Simples



Criando Projeto

Create New Project

 Target Android Devices

Select the form factors your app will run on

Different platforms require separate SDKs

☒ Phone and Tablet

Minimum SDK

API 19: Android 4.4 (KitKat)

Lower API levels target more devices, but have fewer features available. By targeting API 19 and later, your app will run on approximately 33.9% of the devices that are active on the Google Play Store. [Help me choose.](#)

☐ TV

Minimum SDK

API 20+: Android L (Preview)

☒ Wear

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ Glass (Not Installed)

Minimum SDK

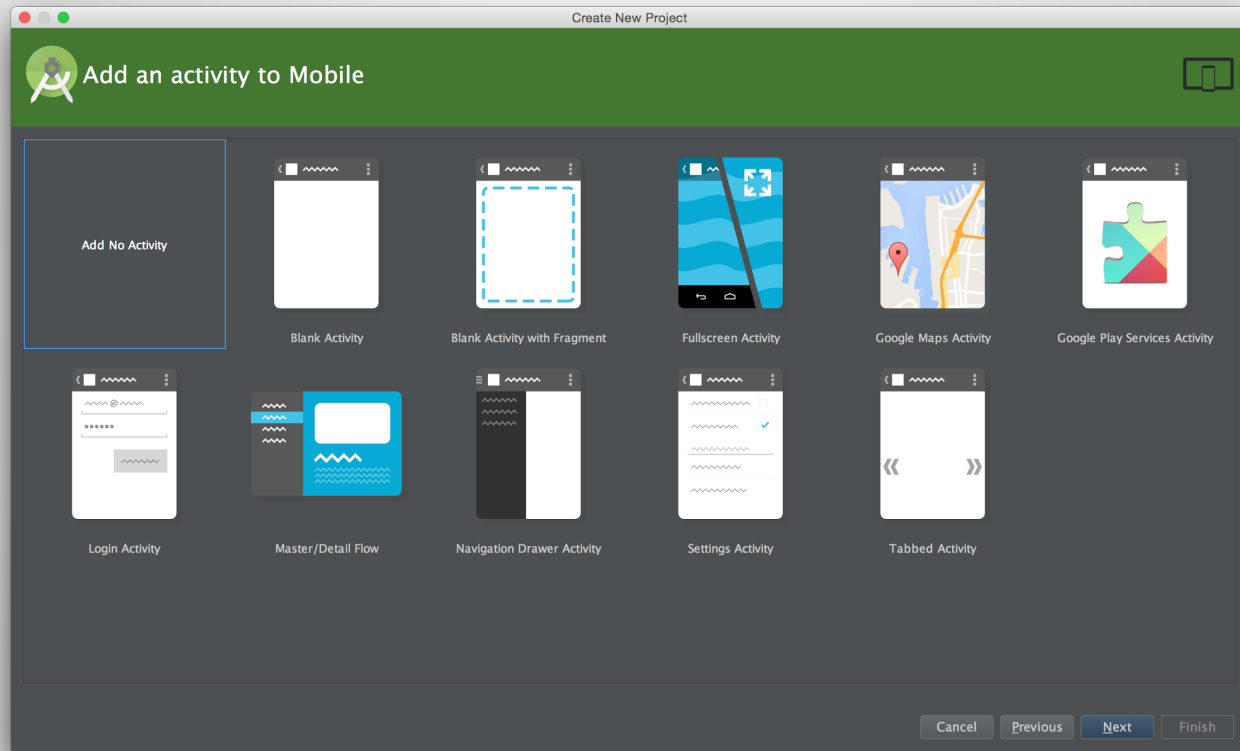
Cancel

Previous

Next

Finish

Criando Projeto



Criando Projeto

Para implementar watchface você vai precisar de dois componentes:

- `CanvasWatchFaceService` - classe base que desenha com `Draw`
- `CanvasWatchFaceService.Engine` - implementação do watchface

Criando Projeto

CanvasWatchFaceService implementa um único método onCreateEngine() que retorna a execução do CanvasWatchFaceService.Engine

Criando Projeto

Antes de executar o projeto, é necessário registrar o serviço no `AndroidManifest.xml` no módulo wear.

Criando Projeto

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.com.rafaelalves.watchfacedigital">
    <uses-feature android:name="android.hardware.type.watch" />
    <uses-permission android:name="com.google.android.permission.PROVIDE_BACKGROUND" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <application
        android:allowBackup="true"
        android:label="WatchFaceDigital"
        android:icon="@mipmap/ic_launcher"
        android:theme="@android:style/Theme.DeviceDefault">

        <service android:name=".SimpleWatchFaceService"
            android:label="WatchFaceDigital"
            android:permission="android.permission.BIND_WALLPAPER">
            <meta-data
                android:name="android.service.wallpaper"
                android:resource="@xml/watch_face" />
            <meta-data
                android:name="com.google.android.wearable.watchface.preview"
                android:resource="@drawable/preview_rectangular" />
            <meta-data
                android:name="com.google.android.wearable.watchface.preview_circular"
                android:resource="@drawable/preview_circular" />
            <meta-data
                android:name="com.google.android.wearable.watchface.companionConfigurationAction"
                android:value="br.com.rafaelalves.watchfacedigital.CONFIG_DIGITAL" />

            <intent-filter>
                <action android:name="android.service.wallpaper.WallpaperService" />
                <category android:name="com.google.android.wearable.watchface.category.WATCH_FACE" />
            </intent-filter>
        </service>

    </application>
</manifest>
```

Criando Projeto

Watch face requer duas permissões:

- `com.google.android.permission.PROVIDE_BACKGROUND`
- `android.permission.WAKE_LOCK`

Implementando Engine

`CanvasWatchFaceService.Engine` fornece o método `onCreate(SurfaceHolder holder)` onde que define o estilo do relógio e outros elementos gráficos.

Engine Callbacks

CanvasWachFaceService.Engine fornece uma série de callbacks que podem ser implementadas.

onDraw(Canvas canvas, Rect bounds)

Provavelmente o mais importante retorno do callback. É chamado cada vez que o relógio é invalidado. Aqui define a lógica de como o relógio vai ser desenhado usando o Canvas e o Rect que define os limites do relógio.

onTimeTick()

Este retorno é invocado a cada minuto quando o relógio está no modo ambiente. Geralmente aqui vamos só chamar `invalidate()` a fim de desencadear `onDraw()`.

onVisibilityChanged(boolean visible)

É chamado quando o relógio se torna visível ou não. Se for sobrecarregar esse método, primeiro deve ser chamado `super.onVisibility(visible)`.

onAmbientModeChanged(boolean inAmbientMode)

Chamado quando o dispositivo entra ou sai do modo de ambiente. Quando em modo ambiente, deve-se estar atento para preservar o consumo de bateria, proporcionando display preto e branco e não ter qualquer animação como a exibição de segundos.

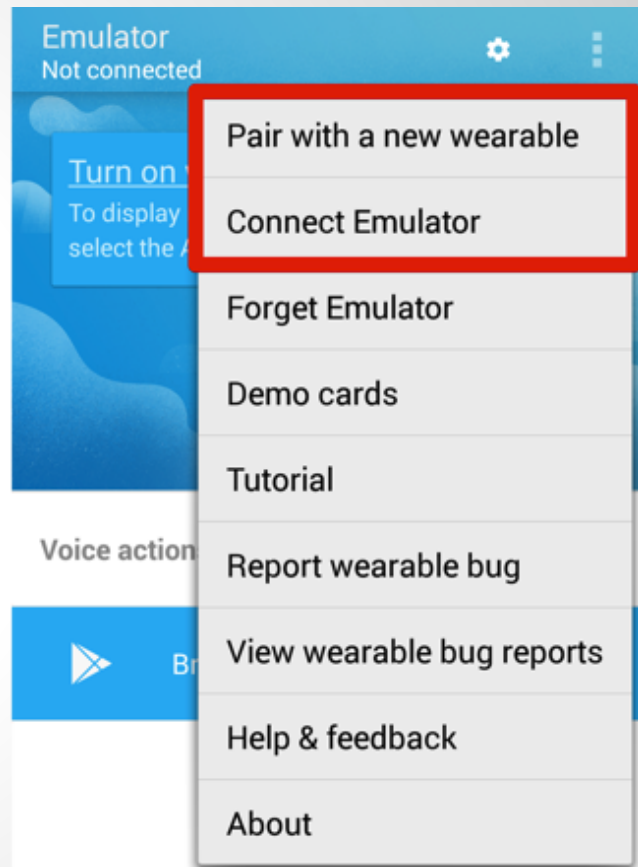
Gerenciando Tempo

Uma das partes mais importante da construção do watch face, é gerenciar adequadamente o tempo. Como vimos, `onTimeTick()` é chamado somente no modo ambiente e a cada minuto. O que fazer para ser notificado a cada minuto se não estiver no modo ambiente? A documentação menciona que você tem que fornecer seu sistema de gerenciamento de tempo.

Executando

1 - Instale o android wear no seu dispositivo.

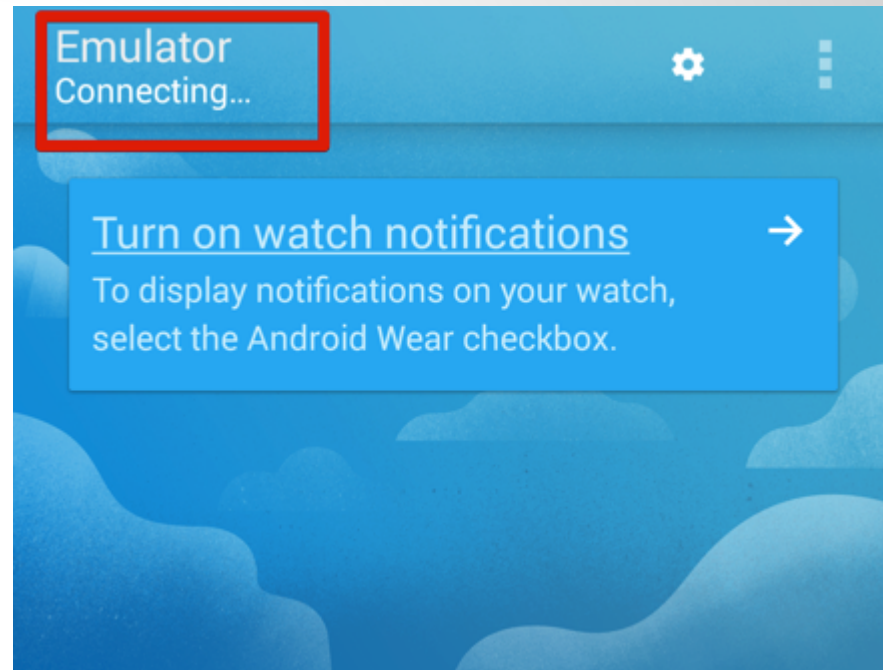
2 - Inicie o aplicativo e conecte-se ao relógio ou emulador.



Executando

3 - (Somente emulador)

Se estiver com problemas na conexão com o emulador, pode ser necessário abrir um canal tcp no seu dispositivo.



Executando

Na linha de comando, acesse sdk/plataform-tools e execute o seguinte comando:

```
adb - d forward tcp:5601 tcp:5601
```

Executando

Ative as opções de desenvolvedor do seu relógio e do seu telefone.

Executando

