
AGENDA 6

WIDGETS



GEEaD - Grupo de Estudos de Educação a Distância

Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO CURSO TÉCNICO EM
DESENVOLVIMENTO DE SISTEMAS PROGRAMAÇÃO MOBILE I

Expediente

Autor:

GUILHERME HENRIQUE GIROLI

Atualização Técnica:

ROGÉRIO GALDIANO DE FREITAS

Revisão Técnica:

Eliana Cristina Nogueira Barion

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim



O que é Widget?

O Widget, como sua tradução já diz, é uma ferramenta encontrada na tela inicial dos sistemas operacionais, para realizar determinadas funções de maneira rápida e ágil. Podemos dizer que ele é uma miniatura de um aplicativo ou programa, onde nessa miniatura contempla algumas das principais funções de um aplicativo ou programa maior.

Essa ferramenta surgiu nos sistemas operacionais para computador, e logo ganhou adeptos pela sua praticidade. Um dos primeiros Widgets desenvolvidos para computadores foi o famoso “bloco de anotações” ou conhecido pela marca “Post-it”. Esse programa é a versão virtual do bom e velho papel amarelo de anotações, que colamos em agendas, cadernos entre outros locais, para registrar informações momentâneas e importantes. A **Figura 1** mostra o programa de anotações que ganhou fãs em todo o mundo.

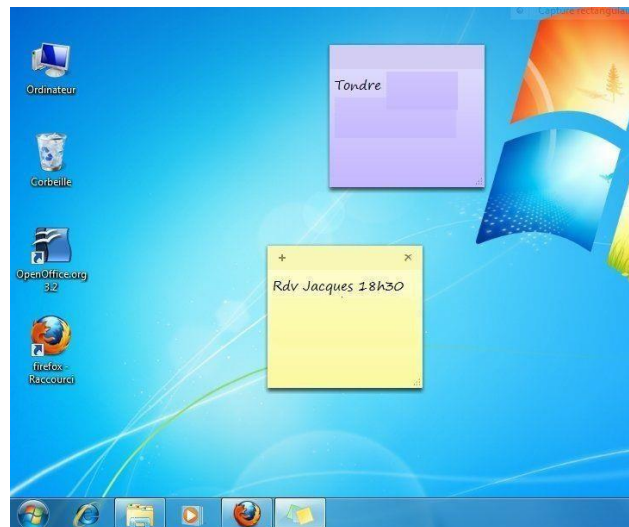


Figura 1 - Widget e anotações para o sistema operacional Windows 7

Como os sistemas operacionais para dispositivos móveis geralmente implementam as ferramentas que tiveram sucesso nos sistemas para computadores, o sistema Android incorporou os Widgets em sua tela inicial, permitindo assim que os desenvolvedores explorem ao máximo esse recurso, proporcionando ótimas experiências de utilização para os usuários em seus aplicativos.

Em um projeto Android, é necessário verificar alguns pontos importantes antes de desenvolver um Widget que trabalhe em conjunto com sua ferramenta principal. Primeiramente é fundamental verificar a real importância do desenvolvimento desta etapa, ou seja, se realmente o projeto necessita de um Widget, como por exemplo, um projeto de calculadora não tem a necessidade de receber uma implementação de um Widget, uma vez que isso só deixaria o projeto mais complexo e pesado para o dispositivo mobile. Em contrapartida, um projeto de aplicativo de vendas de uma loja, poderia contar com um Widget na tela inicial, informando os usuários das principais promoções dessa determinada loja.

Outro ponto importante é escolher qual tipo de Widget seu projeto necessita, a seguir vamos conhecer os principais tipos para o sistema operacional Android.

Widget de informação

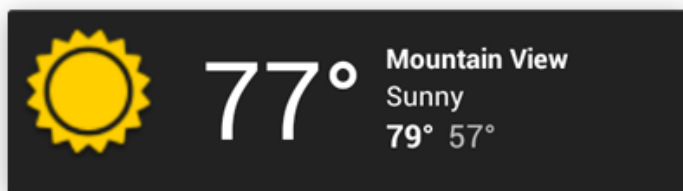


Figura 2 - Widget de informação, sobre previsão do tempo.

Um Widget de informação nada mais é que um elemento na tela do usuário que apresenta informações sobre determinado tema, buscando informações em um banco de dados local ou remoto. Um exemplo simples e de fácil compreensão são os Widgets de previsão do tempo. Eles apresentam em intervalos de tempo atualizações sobre os dados meteorológicos de uma determinada cidade. A figura 2 mostra um Widget de informação.

Widget de coleção

As coleções são formadas através de galerias ou listas, e podemos definir um Widget de coleção pelo fato dele exibir mais de uma informação ao mesmo tempo. Podemos citar um exemplo de um aplicativo de E-mail que possui um Widget para a tela inicial do dispositivo, onde mostra as mensagens da caixa de entrada da conta de E-mail.

Desta forma o Widget pertence a um único aplicativo, e como mostra diversos e-mails ao mesmo tempo, ele é considerado uma coleção. A **Figura 3** mostra um Widget de coleção.

Este tipo de Widget conta com a função de rolagem vertical em seu sistema, podendo exibir muitas informações independente do seu tamanho, e após o usuário escolher qual informação deseja visualizar, apenas uma é aberta por vez.

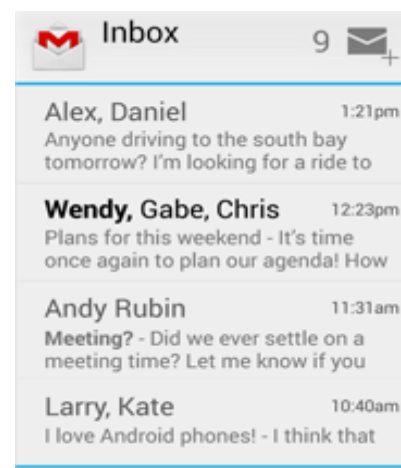


Figura 3 – Widget de coleção, Gmail da Google

Widget de controle

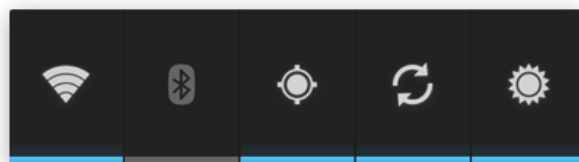


Figura 4 - Widget de controle

Widgets de controle são aqueles utilizados como atalhos na tela, para gerar controle em um aplicativo maior. Geralmente esses Widgets são compostos apenas por botões não exibindo informações em seu contexto.

Um exemplo clássico de Widget de controle são os botões de avançar, voltar e pausar/tocar música de um player de áudio qualquer presente no mercado de aplicativos. Esse Widget fica na tela principal, e facilmente o usuário tem acesso as funções, sem a necessidade abrir o player.

A **Figura 4** mostra um exemplo de Widget de Controle presente na maioria dos celulares, esse conjunto de botões ativa ou desativa alguns recursos do aparelho, sem a necessidade do usuário navegar pelas configurações do sistema operacional.

Widget híbridos

Alguns Widgets podem se enquadrar em mais de um tipo, neste caso ele passa a se chamar Widget híbrido. Um Exemplo fácil e claro é de um player de música, que além de ter os botões de controle, exibe as informações sobre qual a faixa de áudio que está sendo executada. Neste caso ele assumiu as funções de Widget de controle e Widget de informação. A **Figura 5** exibe um exemplo de Widget Híbrido.

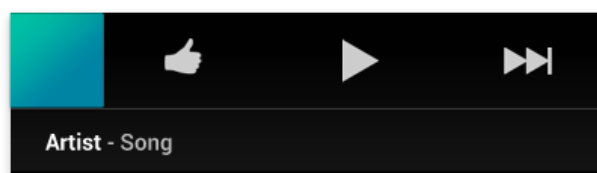


Figura 5 - Widget Híbrido.

Como os Widgets ficam localizados na tela do usuário, independentemente do tipo eles possuem apenas duas formas de gestos utilizadas pelo usuário, ou seja, o usuário apenas pode tocar (clicar na tela) ou deslizar verticalmente (rolar). Essas regras são estabelecidas para que o Widget não atrapalhe outros sistemas, uma vez que o usuário poderia gerar outros toques involuntários na tela tentando aplicar outros gestos no Widget.

No momento a plataforma Kodular, não permite a criação de widgets devido restrições de acesso aos arquivos XML, que são interpretados pelo Sistema Operacional Android, portanto existe a possibilidade de descompilar uma aplicativo (.APK) e realizar as alterações permitindo a construção deste recurso. Mas durante este curso iremos apresentar apenas algumas ferramentas para exibição das informações do dispositivo móvel.

Desenvolvendo um aplicativo

- Abra a plataforma de desenvolvimento do Kodular: <https://www.kodular.io/creator>
- Clique no botão **Create Project**

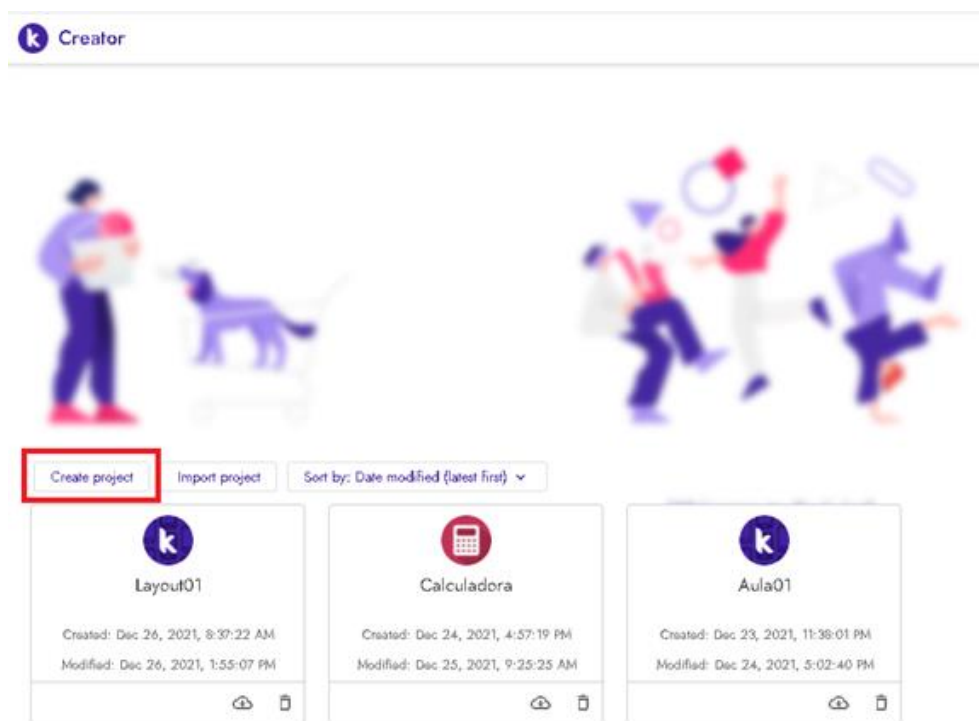
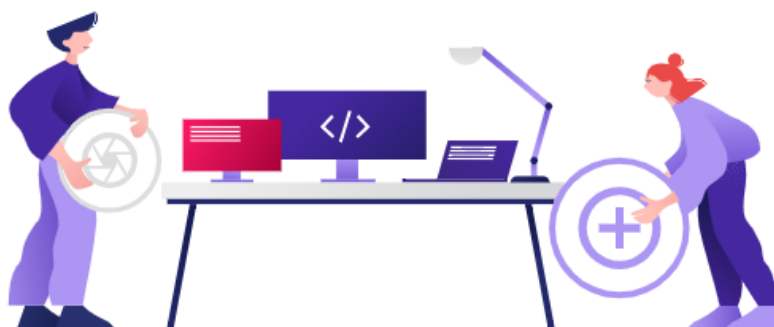


Figura 6 - Criando um novo projeto

- Digite o nome **Utilitario** e clique no botao **Next**.

Create new project



Give your new project a name

Utilitario

Cancel

Next

Figura 7 - Criando o projeto Compartilhar

- Clique no botão **Finish** para finalizar a criação do novo projeto.
- Altere as propriedades do objeto **SCREEN**

Propriedade	Valor	Função
Background Color	#000000FF	Definir a cor de fundo do aplicativo.
Title	Compartilhar Fotos	Definir o título da aplicação em desenvolvimento.
Align Horizontal	Center	Alinhar todos os componentes ao centro.

- Escolha um ícone para sua aplicação, sendo que a padronização do ícone deverá ser feita pelo botão **Configurações**
- Clique na categoria **Layout** na **Palette**.

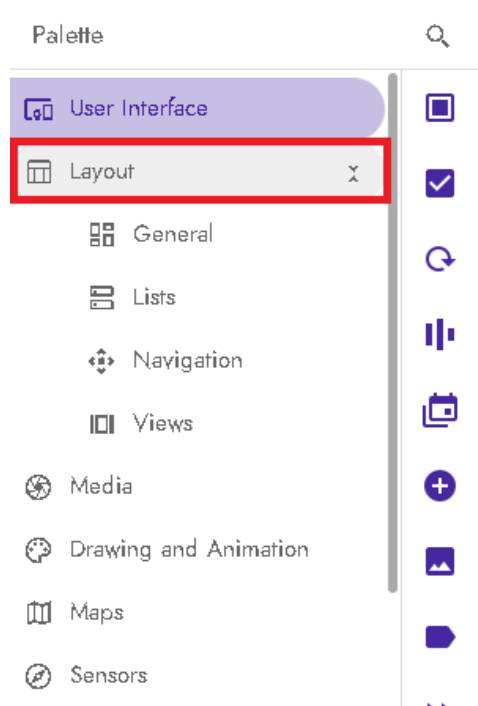


Figura 8 - Criando um novo Layout.

- Clique na Categoria **General**.

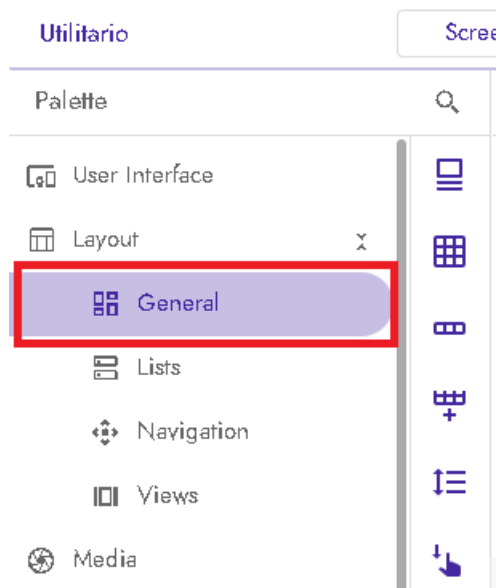


Figura 9 - Opções de Layout

- Clique no componente **HorizontalArrangement** e arraste para opção **VIEWER**.

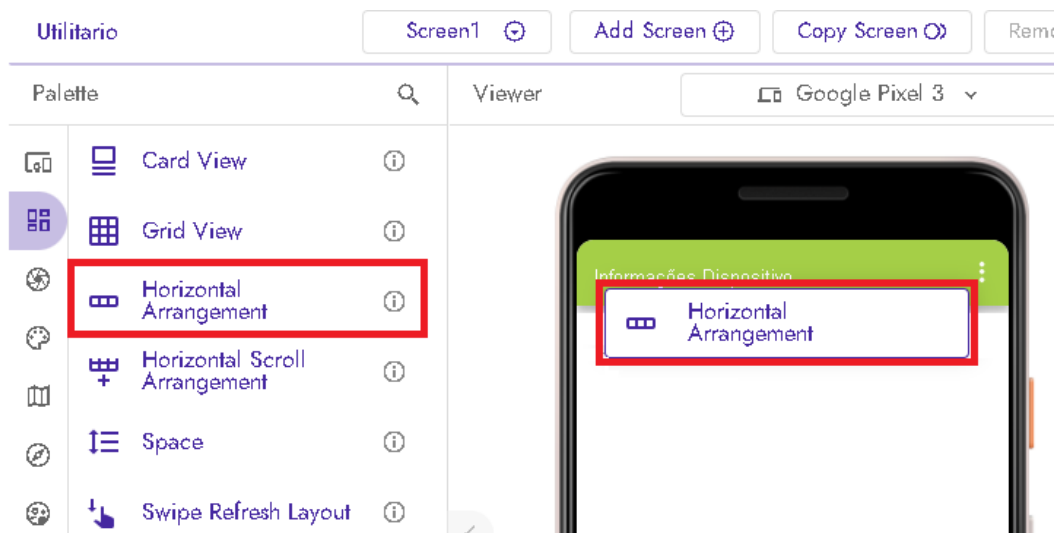


Figura 10 - Inserindo componente HorizontalArrangement

- Altere as propriedades do componente **HorizontalArrangement**, conforme a tabela abaixo:

Propriedade	Valor	Função
Align Vertical	Center	Definir todos os objetos alinhados no centro na vertical.
Width	Fill Parent	A largura do objeto será a mesma largura da tela do dispositivo móvel.
Name	Horizontal_Bateria	Definir o nome do componente

- Retorne para a categoria **User Interface**, selecione o componente **Label** e arraste dentro do componente **Horizontal_Bateria** na área **VIEWER**.

- Alterar as propriedades do **Label** de acordo com a tabela abaixo:

Propriedade	Valor	Função
Font Size	18	Definir o tamanho da fonte que será utilizada no componente.
Font Bold	Marcado	Acrescente o negrito na fonte do componente.
Text	Número 1:	Definir qual o texto que será exibido pelo componente Label.
Name	lbl_Bateria	Definir o nome do componente.

- Inserir o componente **Space** da categoria **Layout**, categoria **General** na área de **VIEWER**.

- Altere as propriedades do objeto **Space**

Propriedade	Valor	Função
Height	40px	Definir a altura em px para representar o espaçamento.
Name	spc_Linha01	Definir o nome do componente

- Inserir o componente **Vertical Arrangement** da categoria **Layout**, categoria **General** na área de **VIEWER**.

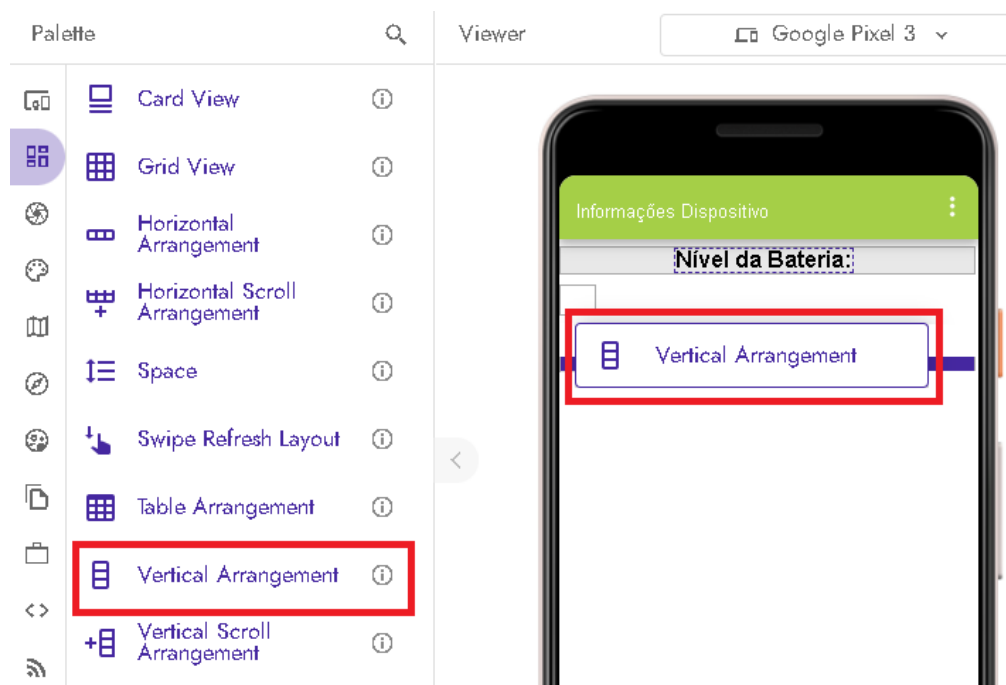


Figura 11 - Inserindo componente Vertical Arrangement

- Altere as propriedades do objeto **Vertical Arrangement**.

Propriedade	Valor	Função
Align Horizontal	Center	Definir o alinhamento horizontal ao centro.
Width	Fill Parent	A largura do objeto será a mesma largura da tela do dispositivo móvel.
Name	Vertical_Dados	Definir o nome do componente

- Clique no componente **Label** e insira na área de **VIEWER**, dentro componente **Vertical_Dados**
- Altere as propriedades do objeto **Label**.

Propriedade	Valor	Função
Font Size	16	Definir o tamanho da fonte.
Font Bold	✓ Marcado	Definir a opção de negrito na fonte
Text	Versão	Definir o texto que será apresentado pelo Label .
Name	lbl_Versao	Definir o nome do componente.

- Clique no componente **Label** e insira na área de **VIEWER**, dentro componente **Vertical_Dados** abaixo do componente **lbl_Versao**
- Altere as propriedades do objeto **Label**.

Propriedade	Valor	Função
Font Size	16	Definir o tamanho da fonte.
Font Bold	✓ Marcado	Definir a opção de negrito na fonte
Text	Idioma	Definir o texto que será apresentado pelo Label .
Name	lbl_Idioma	Definir o nome do componente.

- Clique no componente **Label** e insira na área de **VIEWER**, dentro componente **Vertical_Dados** abaixo do componente **lbl_Idioma**
- Altere as propriedades do objeto **Label**.

Propriedade	Valor	Função
Font Size	16	Definir o tamanho da fonte.
Font Bold	✓ Marcado	Definir a opção de negrito na fonte
Text	Fabricante	Definir o texto que será apresentado pelo Label .
Name	lbl_Fabricante	Definir o nome do componente.

- Clique no componente **Label** e insira na área de **VIEWER**, dentro componente **Vertical_Dados** abaixo do componente **lbl_Fabricante**
- Altere as propriedades do objeto **Label**.

Propriedade	Valor	Função
Font Size	16	Definir o tamanho da fonte.
Font Bold	✓ Marcado	Definir a opção de negrito na fonte
Text	Modelo	Definir o texto que será apresentado pelo Label .
Name	lbl_Modelo	Definir o nome do componente.

Após a inserção de todos os componentes e alterados as respectivas propriedades, o layout deverá ser idêntico a **Figura 12**.



Figura 12 - Visualizando o layout do projeto.

- Clique no componente **HorizontalArrangement**, na categoria **Layout, General** e arraste para opção **VIEWER**.
- Altere as propriedades do componente **HorizontalArrangement**, conforme a tabela abaixo:

Propriedade	Valor	Função
Align Horizontal	Right	Definir todos os objetos alinhados a direita na horizontal.
Align Vertical	Bottom	Definir todos os objetos alinhados na borda inferior do componente.
Height	Fill Parent	A altura do objeto será a mesma largura da tela do dispositivo móvel.
Width	Fill Parent	A largura do objeto será a mesma largura da tela do dispositivo móvel.
Name	Horizontal_Botoes	Definir o nome do componente

- Insira um componente **Button**, categoria **User Interface** dentro do componente **Horizontal_Botoes**.
- Altere as propriedades do componente **Button**.

Propriedade	Valor	Função
Font Bold	✓ Marcado	Definir a opção negrito
Font Size	18	Definir o tamanho da fonte
Width	100px	Definir a largura do componente
Text	Sair	Definir o conteúdo a ser exibido do componente button.
Text Alignment	Center	Definir o alinhamento ao centro.
Name	btn_Sair	Definir o nome do componente.

- Clique na categoria **Utilities**

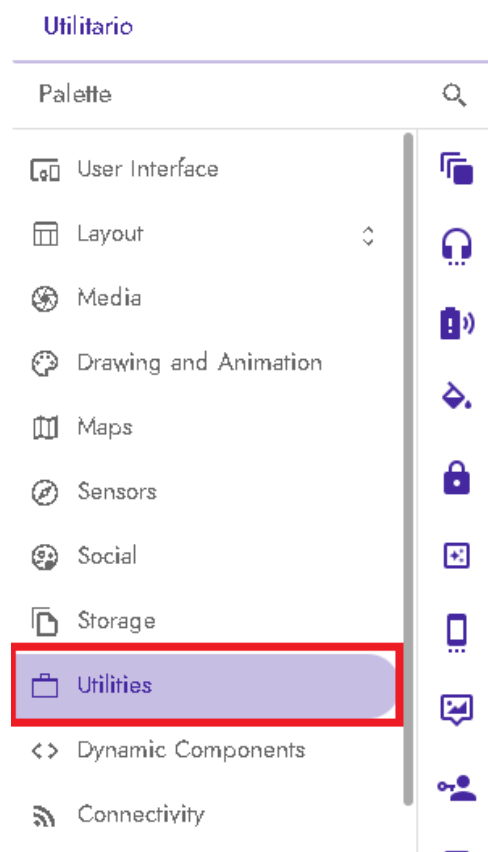


Figura 13 - Categoria Utilities

- Clique no componente **Battery Utilities** e arraste para a área **Viewer**

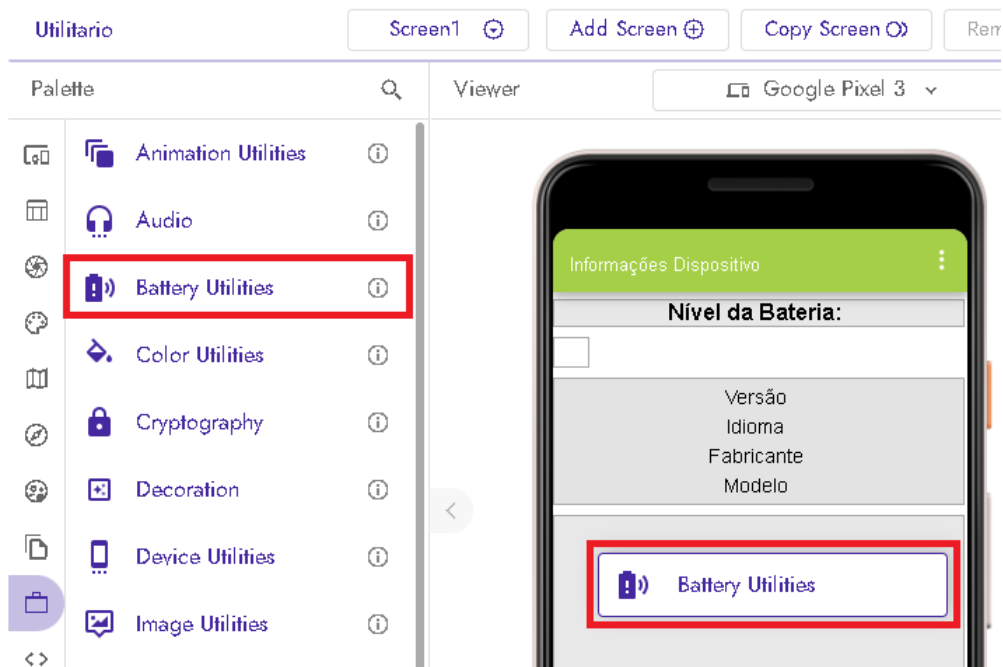


Figura 14 - Inserindo componente Battery Utilities.

- Altere as propriedades do componente **Battery Utilities**.

Propriedade	Valor	Função
Enabled	✓ Marcado	Definir o componente como ativado.
Name	Battery_Util	Definir a opção negrito

- Clique no componente **Device Utilities** e arraste para a área **Viewer**

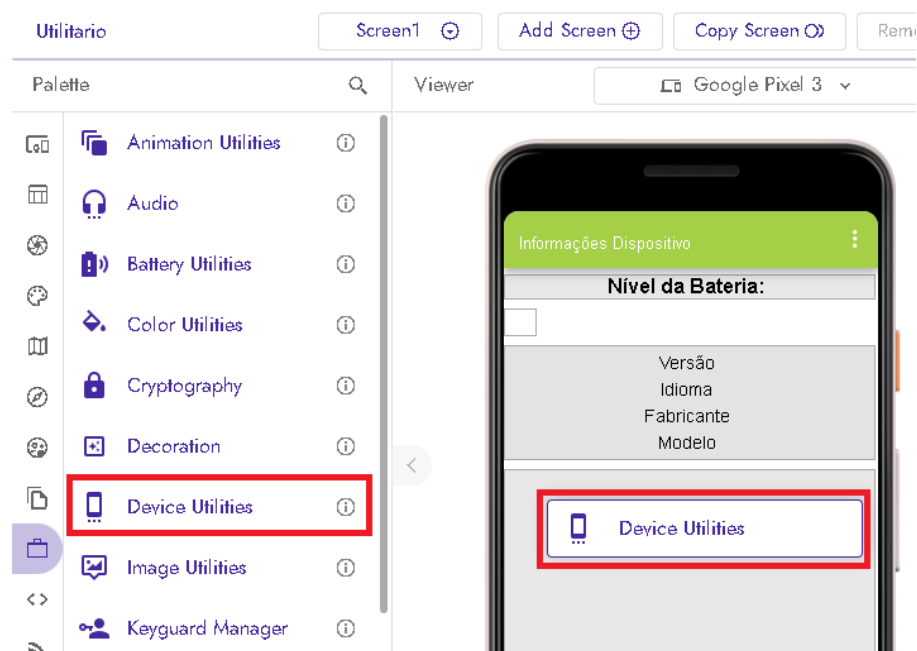
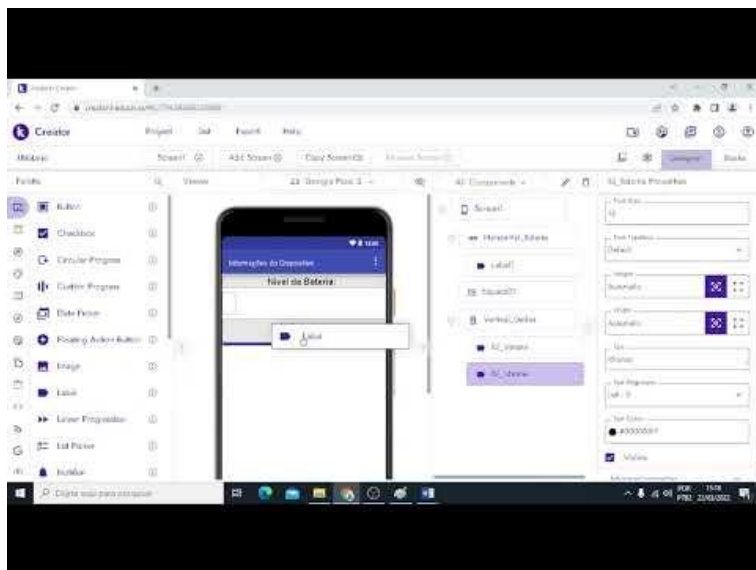


Figura 15 - Inserindo o componente Device Utilities

- Altere as propriedades do componente **Battery Utilities**.

Propriedade	Valor	Função
Enabled	✓ Marcado	Definir o componente como ativado.
Name	Device_Util	Definir a opção negrito

Caso o aluno tenha alguma dúvida na construção do layout deste aplicativo, poderá assistir ao vídeo **Agenda 06 – Criando a interface Utilitário**, disponível em: https://youtu.be/9i767_LDG1s.



Após a construção do layout do nosso aplicativo, vamos alterar para a opção de programação em blocos (Blocks).

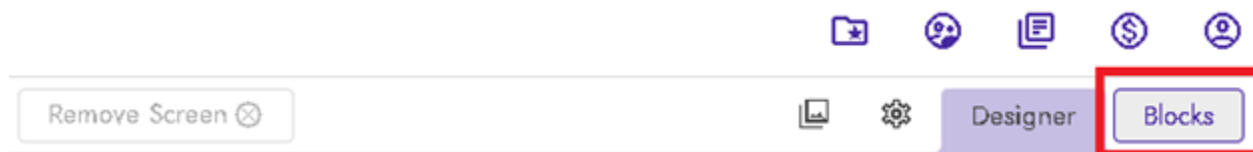


Figura 16 - Alterando a programação em blocos

Vamos criar os blocos de código, conforme a **Figura 17**.

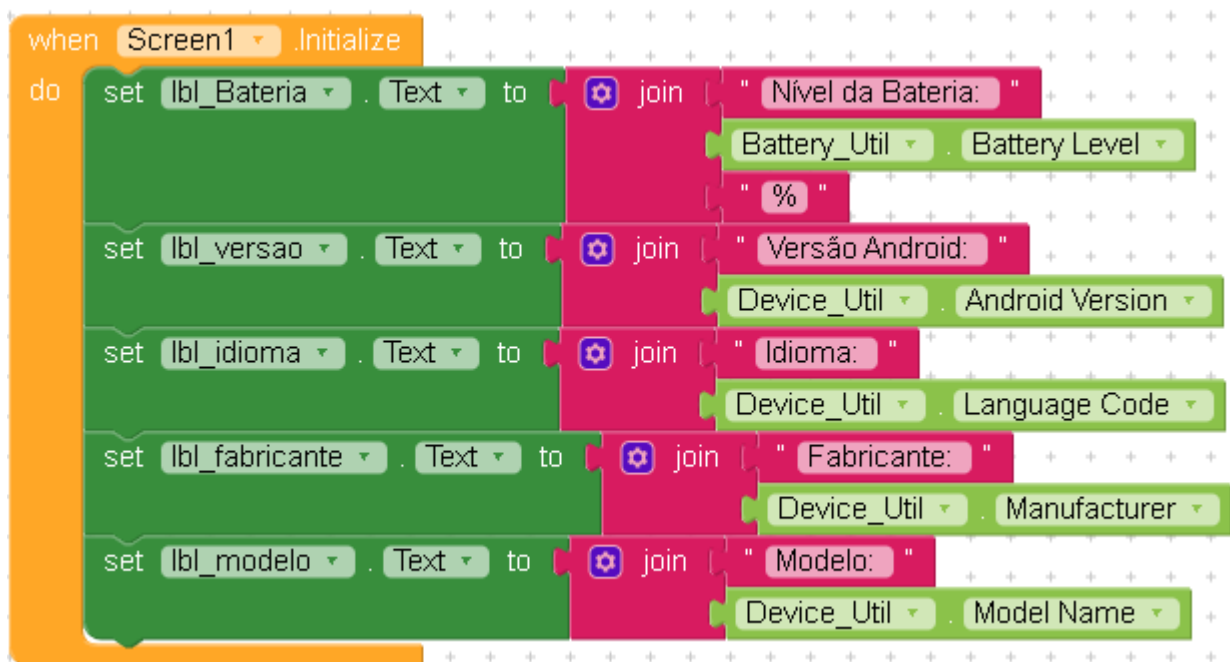
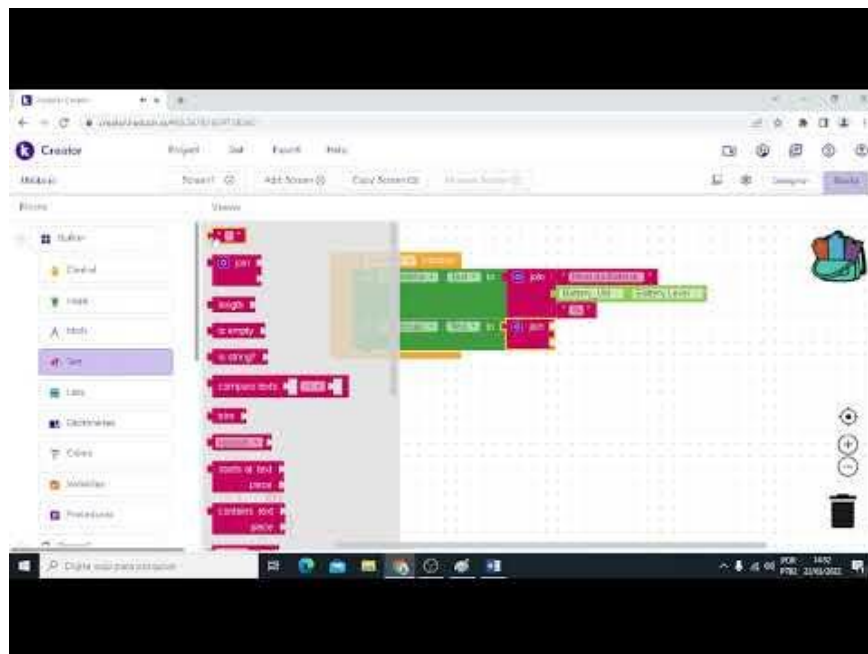


Figura 17 - Blocos de código da programação.

Para solucionar as dúvidas, assista ao vídeo **Agenda 06 - Criando a programação de blocos Utilitário**, disponível em: <https://youtu.be/92tQTqkN52E>



Para finalizar o projeto, vamos exportar o **arquivo APK** para o dispositivo móvel e realizar a instalação através do aplicativo **Kodular Companion**.

- Clique no menu **Export**, na opção Android App (.apk).

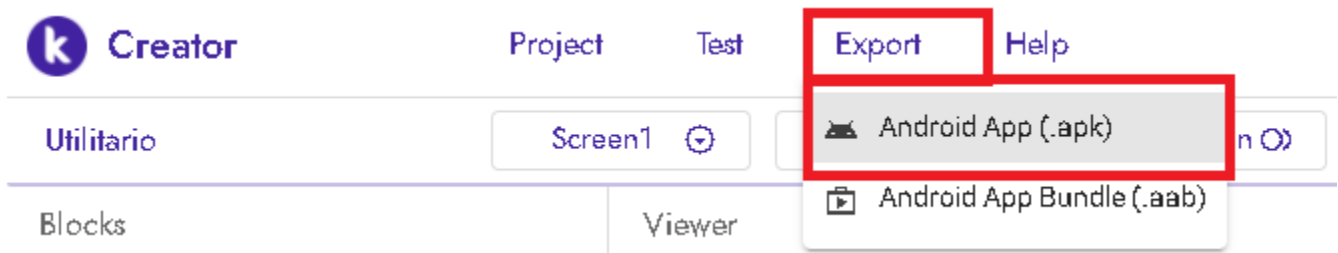


Figura 18 - Menu Export, opção Android App (.apk)

- Utilize o aplicativo **Kodular Companion** para escanear o qrcode e siga todos os passos para a instalação do aplicativo, de acordo com material anterior.

Android App for "Utilitario"

Scan the QR code on your phone to install the app or download the APK file to your computer.

Note: This link is valid only for 10 minutes. It is recommended to export your app as an Android App Bundle for distribution via Google Play.

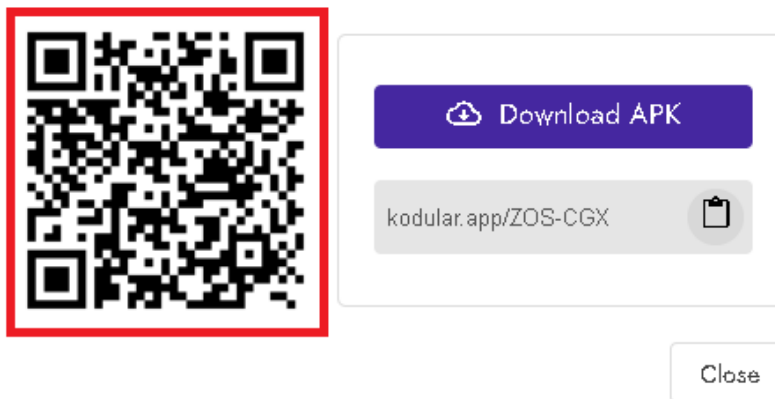


Figura 19 - Disponibilizando o Qrcode