
AGENDA 1

TRABALHANDO COM
ÁUDIO E UTILIZANDO
O ATRIBUTO ONCLICK
NO BUTTON



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

Atributo onClick

Verificamos até aqui que para gerar uma “ação” para um determinado botão ou Button era necessário desenvolver uma codificação em blocos, utilizando o evento “onclick” responsável por gerenciar esse botão. Sendo um formato padrão utilizado na plataforma **Kodular** para tratamento do clique do botão, que também é amplamente utilizado pelos desenvolvedores de aplicativos.

O evento “onClick” é declarado na linguagem de programação em blocos. Cabe a ele chamar um método da codificação que, por sua vez, desenvolve uma ação quando o botão é pressionado.

A **Figura 1** demonstra um aplicativo de exemplo com dois botões.

Ambos os botões foram desenvolvidos para serem executados a partir do momento que o usuário clicar sobre eles. Sendo que o responsável para acionar a programação é o evento onclick.

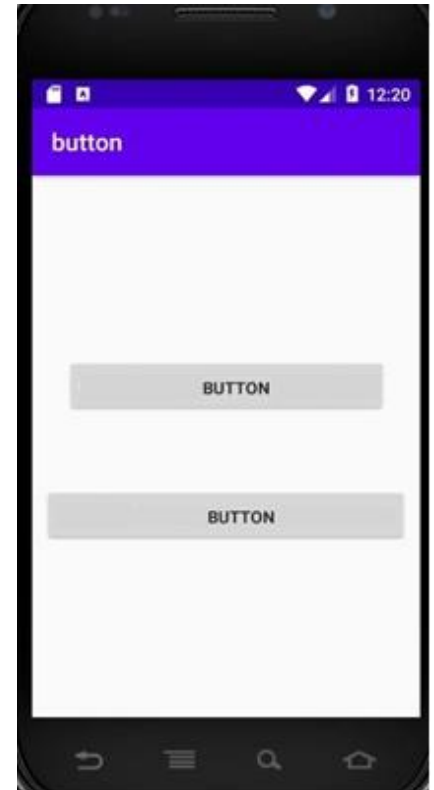


Figura 1 - Exemplo de aplicativo com dois botões

Podemos observar que o sistema por meio do “onClick” oferece uma construção com menos códigos no arquivo quando comparado a outras linguagens de programações”.

Já o atributo “onClick” desenvolve automaticamente, nos “bastidores” da aplicação, uma estrutura para “ouvir” e tratar os cliques do usuário, transferindo apenas para a programação orientada a objetos a função de desempenhar uma ação para esse clique, por meio do método associado a ele.

Portanto, quando o desenvolvedor necessita interferir ou tratar alguns aspectos deste clique antes de desempenhar uma ação do botão. Nos casos mais simples, como no projeto do jogo “Descubra o número...”, em que o desenvolvedor necessita apenas executar uma ação no clique do botão, é aconselhável a utilização do “onClick”, que ocupa uma quantidade menor de linhas de código na programação, onde é necessário apenas a construção de um evento.

Componente Sound

O componente Sound é especializado na reprodução de sons. Quando a classe é utilizada em um projeto, proporciona, de maneira fácil, a execução de áudio, vídeo e imagens no aplicativo.

O componente Soundr permite reprodução de áudio de arquivos de mídia armazenados nos recursos do aplicativo, de arquivos na memória do dispositivo ou de um fluxo de dados por meio da Internet.

Com poucos códigos, é possível desenvolver uma ferramenta capaz de executar arquivos dos mais diversos tipos, entre eles MP3, MP4, WAV, entre outros..

O código a seguir apresenta um exemplo que implementa um objeto utilizando o componente Sound.



Figura 2 - Componente Sound

Quando declaramos um componente Sound, utilizamos o enveto **play** para informar algumas coisas importantes para a execução do áudio. A primeira parte do componente sound é destinada à informação do contexto da aplicação, onde vamos informar a tela que vai executar o áudio. Na segunda parte, informamos a localização do arquivo. O arquivo de áudio está disponível na pasta de upload do aplicativo.



Antes de começar a codificar o aplicativo em **Kodular** é importante lembrar que essa apostila é uma continuação do projeto de um jogo com nome “Descubra o número...” desenvolvido por Karla. Todo o processo de desenvolvimento da parte visual do projeto foi realizado na agenda anterior do módulo anterior, caso você tenha perdido algum detalhe, volte no material anterior e execute os procedimentos. Vamos tratar a partir daqui apenas a parte que envolve a programação em blocos.

Random

A classe **Random** é utilizada para gerar valores aleatórios em um aplicativo Android. Essa classe utiliza alguns métodos de acordo com a necessidade do desenvolvedor, no jogo será necessário utilizar um valor aleatório do tipo inteiro em cada partida. Neste caso, é necessário utilizar o método **random integer from**.

O **random integer from** é responsável por gerar valores inteiros de 1 (um) até um determinado limite. Para definir o limite, o método utiliza um parâmetro que define o valor máximo que será gerado aleatoriamente, este valor máximo informado nunca será gerado pelo sistema. No exemplo de código a seguir, definimos um gerador de números inteiros de 1 (um) até 10 (dez), que alimenta uma variável do tipo inteiro. Lembrando que o objeto **random integer from** pertence ao construtor **Math**.



Figura 3 - Gerando um número randômico.

Portanto vamos lembrar qual é a interface deste aplicativo. Sendo que o usuário deverá clicar no botão **Sortear** para que o aplicativo possa gerar um número aleatório. Assim o usuário irá digitar o nome que acha que foi gerado no campo **txt_numero** e clicar no botão **Enviar**, caso o número digitado seja o número correto o aplicativo irá emitir um som de aplauso. Caso contrário, o usuário poderá clicar no botão **Sortear** e o aplicativo irá gerar outro número randômico.

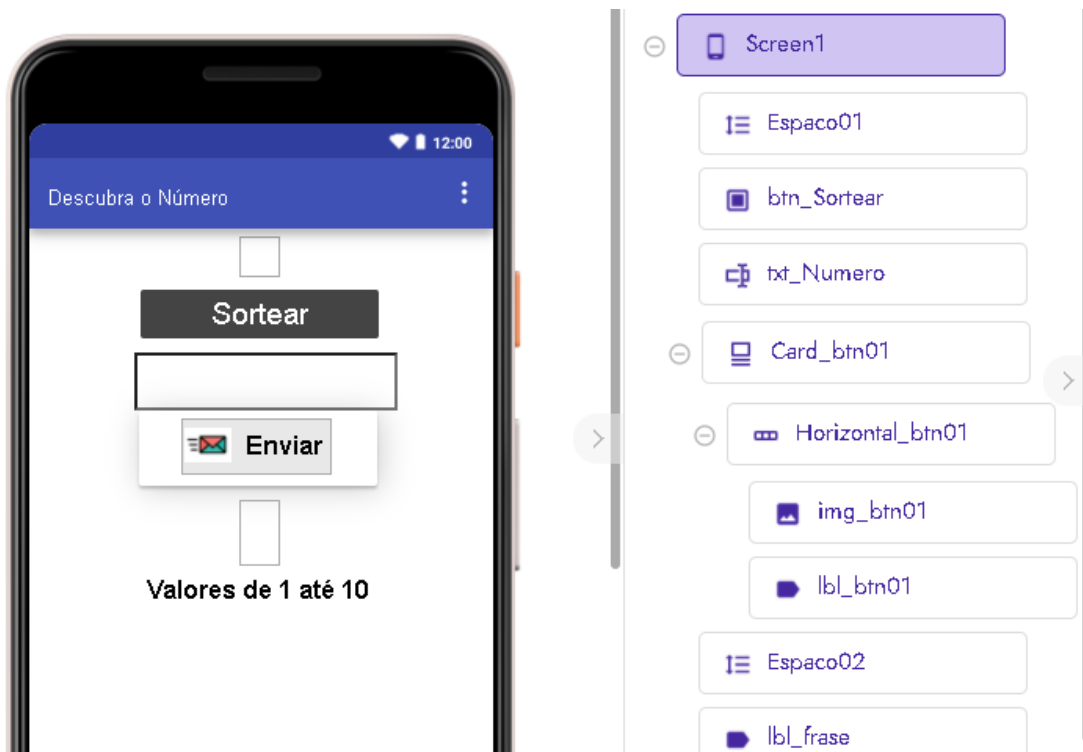


Figura 4 - Visualizando o layout

- Clique na categoria **Media**.

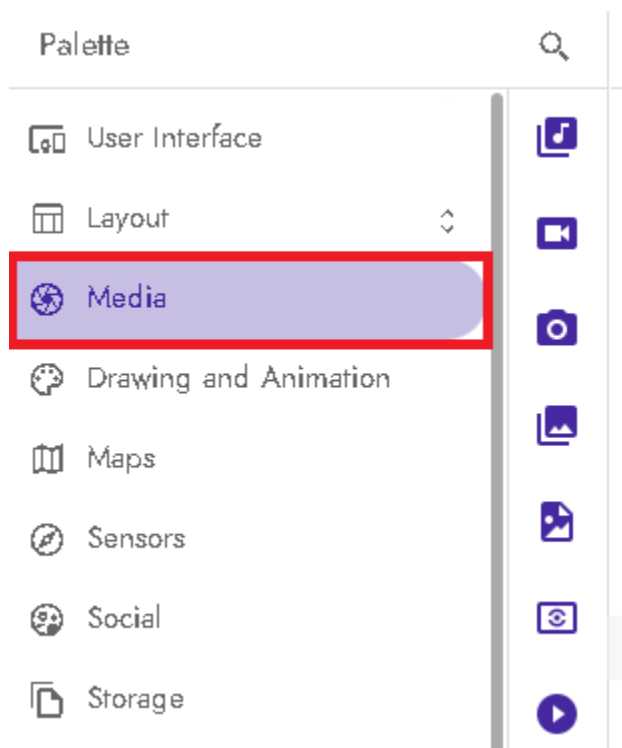


Figura 5 - Categoria Média

- Clique no componente **Sound** e arraste para área **VIEWER**.

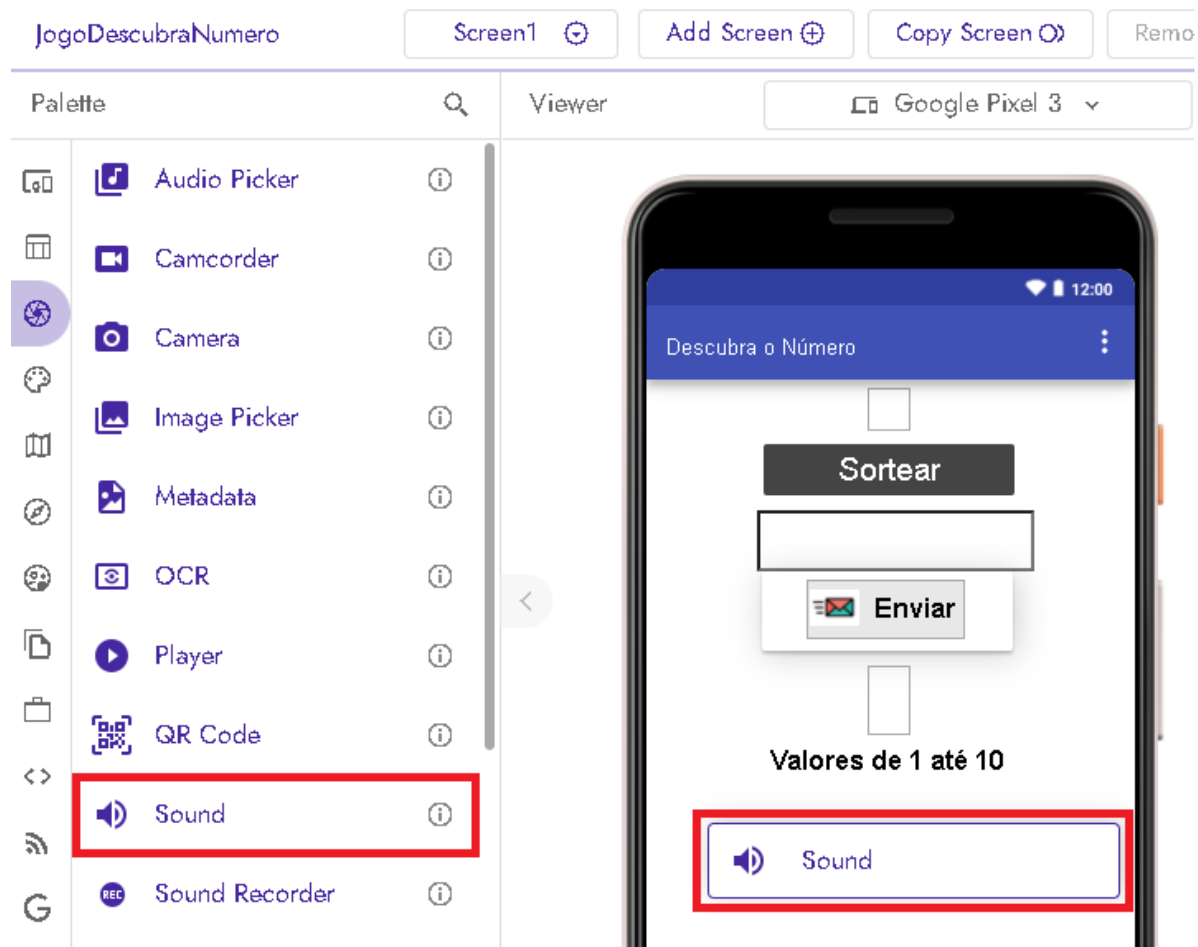


Figura 6 - Inserindo o componente Sound.

- Renomear o componente **Sound1** para o nome de **Som**.

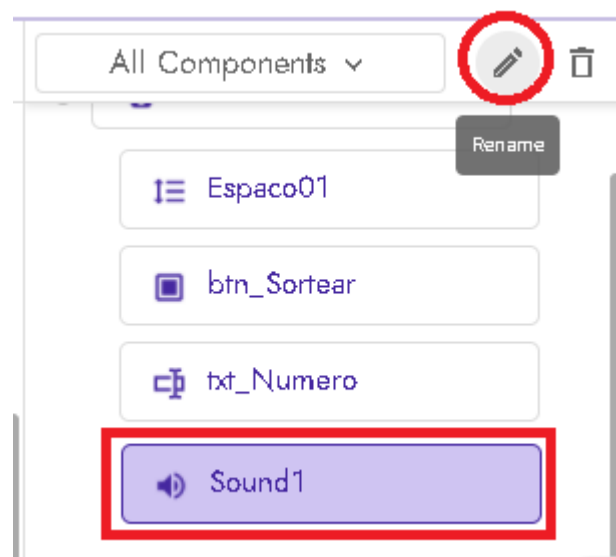


Figura 7 - Renomear o componente.

- Clique no componente **Som** e realizar o upload do arquivo **aplausos.mp3** .

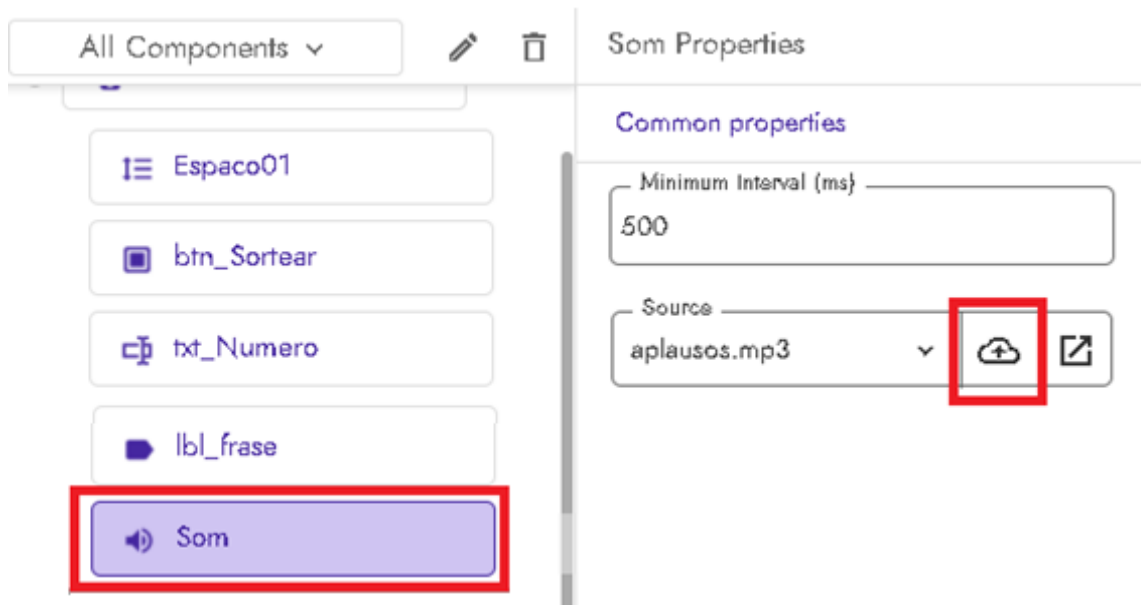


Figura 8 - Upload do arquivo aplausos.mp3

- Clique na categoria **User Interface**, clique no componente **Notifier** e arraste para a área **VIEWER**.

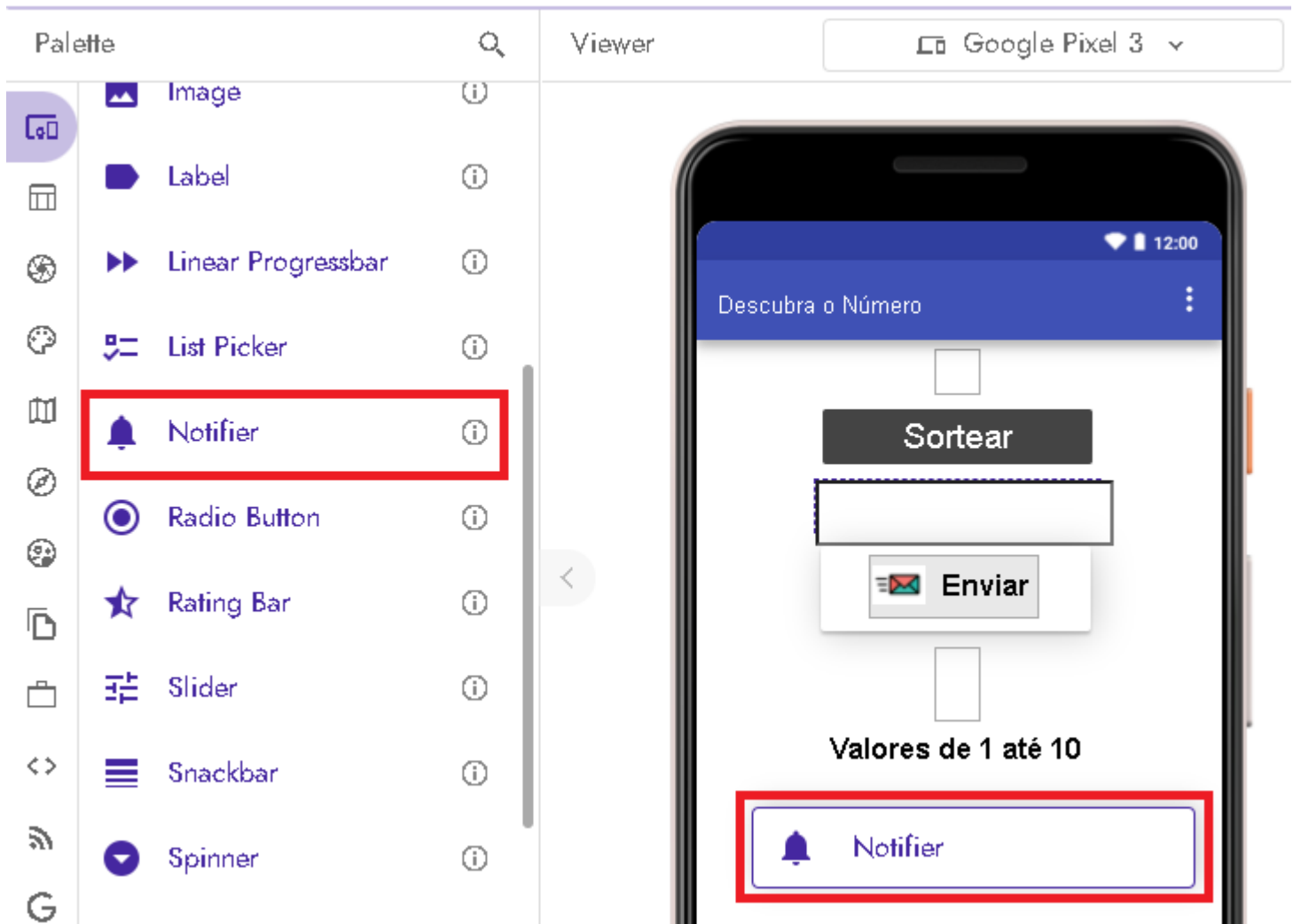


Figura 9 - Inserindo o componente Notifier

A função deste componente será apresentar notificações, ou seja, apresentar uma caixa de diálogo para exibir informações importantes.

- Altere as propriedades do componente **Notifier**.

Propriedade	Valor	Função
Name	Notifica	Definir o nome do componente.

Chega de conversa, vamos programar!

Construa o código de acordo com a **Figura 10**, para que o aplicativo realize todas as funções necessárias.

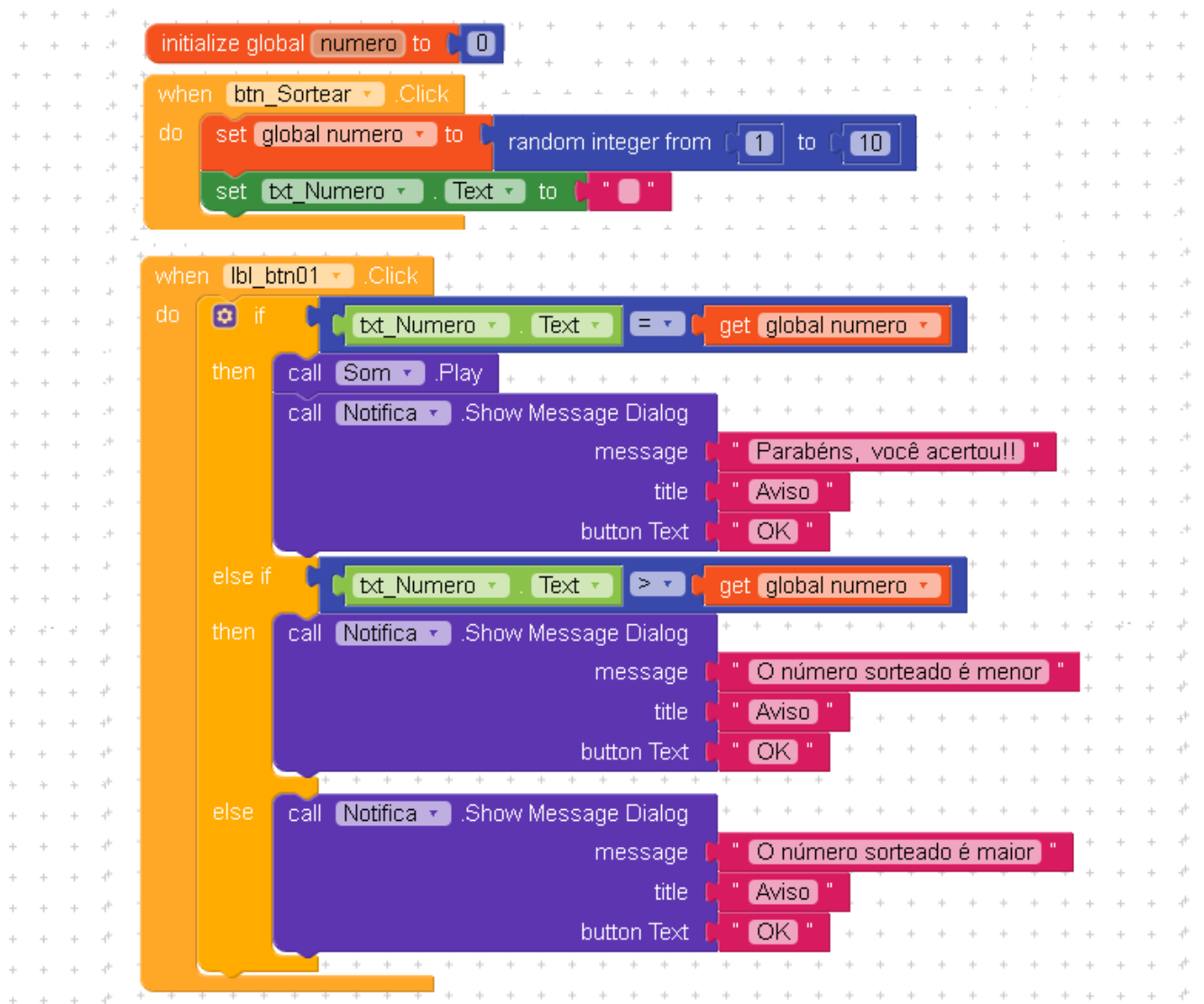
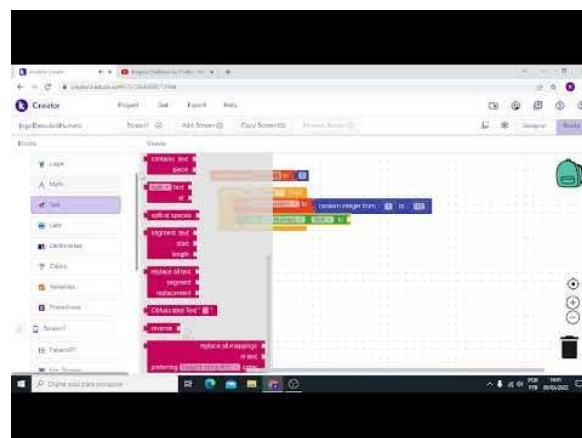


Figura 10 - Programação em blocos completa do aplicativo.

Caso tenha alguma dúvida em relação a criação do blocos de programação, poderá assistir ao vídeo:

Agenda 01 – Criando a programação do JogoDescubra Número, disponível em:

<https://youtu.be/23aawaQQNTw>



Para finalizar o projeto, o aluno deverá 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).

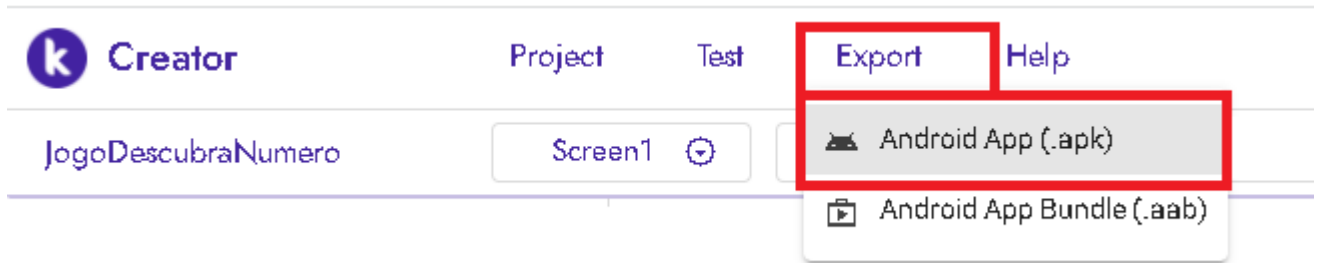


Figura 11 - Menu Export, opção Android App (.apk)

- 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 "JogoDescubraNumero"

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.

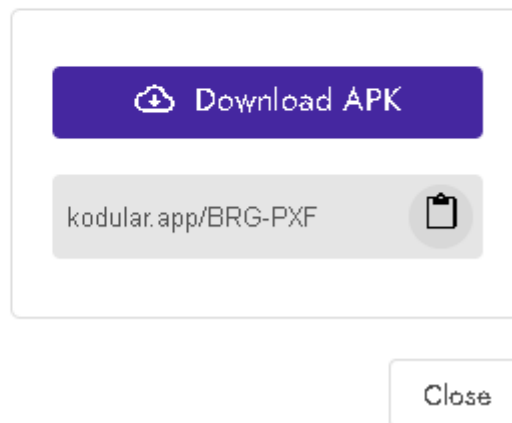


Figura 12 - Disponibilizando o Qrcode.



Envie para o seu tutor um vídeo mostrando a tela do celular com o projeto realizado por você na etapa "Você no comando". Envie também os códigos utilizados. Aproveite para customizar o projeto **"JogoDescubraNumero"** com suas cores e uma imagem escolhida por você.

Arquivos disponíveis para download

[Projeto Finalizado](#)

[Imagens utilizados no projeto JogoDescubraNumero.aia](#)

[Som utilizado no projeto JogoDescubraNumero.aia](#)