AGENDA 5

PERMISSÕES

GEEaD - Grupo de Estudos de Educação a DistânciaCentro de Educação Tecnológica Paula Souza

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

Expediente

Autor:

GUILHERME HENRIQUE GIROLLI

Atualização Técnica:

Rogério Galdiano de Freitas

Revisão Técnica:

Eliana Cristina Nogueira

BarionRevisão

Gramatical:

Juçara Maria Montenegro Simonsen

SantosEditoração e Diagramação:

Flávio Biazim

São Paulo – SP, 2021

MERGULHANDO NO TEMA...

Com a evolução na aprendizagem do Kodular, nos deparamos na agenda anterior com uma situação bem peculiar, onde o aplicativo desenvolvido por Serginho consumia um recurso do dispositivo Mobile, ou seja, o aplicativo tinha que conectar na Internet para adquirir as informações de um servidor JSON. Essa situação foi barrada pelo sistema operacional Android, e para que o aplicativo funcionasse de maneira correta, foi preciso efetuar uma liberação de acesso desse recurso.

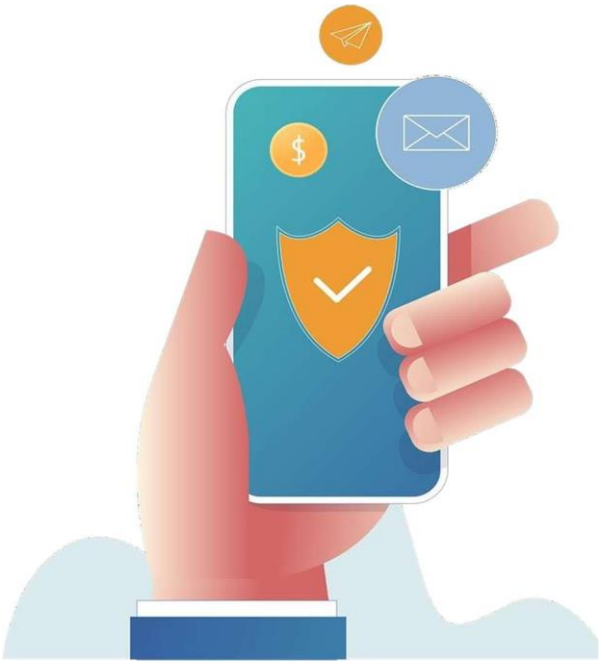


Figura 1 - Conceitos de Permissões

Você deve estar se perguntando: por qual motivo o Android barrou a utilização da Internet para o aplicativo de Serginho? E a resposta é muito simples e objetiva: - Segurança para o usuário do dispositivo Mobile!

Imagine que o aplicativo desenvolvido para consultar CEP, seja instalado em um dispositivo Mobile que possui uma franquia pequena para utilização dos recursos de Internet de uma operadora qualquer de telefonia celular. Nesse cenário, se o sistema operacional Android não se preocupar em limitar acessos a Internet para o aplicativo, alguns problemas poderiam surgir, como até mesmo cobranças financeiras para o usuário. Desta forma nada melhor que avisar, ou seja, deixar o usuário do aplicativo ciente de que ele utiliza recursos de Internet.

Esse foi um exemplo bem simples e menos complexo, porém o desenvolvimento de aplicativos Mobile lida

com outros recursos mais complexos e bem mais preocupantes dependendo do projeto. Alguns exemplos são o acesso a informações particulares como fotos e vídeos do usuário, acesso a informações do cartão de memória de um dispositivo, câmera, microfone etc. Se esses recursos não forem controlados pelo sistema operacional, e sua liberação de utilização for liberada pelos usuários, podem surgir problemas graves, tornando o sistema falho e de baixa segurança, o que seria um paraíso para pessoas má intencionadas e para os criminosos digitais!

Nas últimas agendas, estudamos que o sistema operacional Android foi concebido por meio do Kernel do sistema Linux, portanto, ele traz em seu DNA uma política rigorosa de segurança que garante alta proteção aos seus usuários.

O desenvolvedor de aplicativos para esse sistema operacional obrigatoriamente tem que conhecer algumas regras básicas, à medida que vai se aprofundando na programação Mobile.

A regra primordial é que o sistema operacional possua um conjunto de permissões para que o aplicativo acesse e/ou utilize determinadas áreas do dispositivo. Por padrão, todas as permissões são bloqueadas, desta forma não corre o risco de um aplicativo prejudicar o outro ou até mesmo o sistema operacional do aparelho, sem ter que passar por uma autorização prévia.

Como todo aplicativo é executado separadamente um do outro, as permissões são exclusivas, ou seja, o que é autorizado para um aplicativo, não será autorizado, de maneira automática, para os demais.

O desenvolvedor Mobile não deve se preocupar com permissões para projetos básicos e aplicativos simples, pois os recursos básicos de processamento e execução não oferecem risco para o sistema e são liberados para o usuário.

Os recursos são divididos basicamente em **“Permissões Normais”** e **“Permissões Perigosas”.**

Permissões Normais

As permissões normais são aquelas que o aplicativo utiliza e que oferecem um pequeno risco à privacidade do usuário ou ao sistema operacional. Essas permissões ocorrem durante a execução do aplicativo e o sistema operacional concede a devida autorização automaticamente, sem a necessidade de informar o usuário. Um exemplo é a utilização da Internet por um aplicativo de rede social.

Figura 2 - Permissões

Para conhecer as principais "Permissões Normais" verifique a figura 3, retirada do site oficial do sistema Android.

- ACCESS_LOCATION_EXTRA_COMMANDS
- ACCESS_NETWORK_STATE
- ACCESS_NOTIFICATION_POLICY
- ACCESS_WIFI_STATE
- BLUETOOTH
- BLUETOOTH_ADMIN
- BROADCAST_STICKY
- CHANGE_NETWORK_STATE
- CHANGE_WIFI_MULTICAST_STATE
- CHANGE_WIFI_STATE
- DISABLE_KEYGUARD
- EXPAND_STATUS_BAR
- FOREGROUND_SERVICE
- GET_PACKAGE_SIZE
- INSTALL_SHORTCUT
- INTERNET
- KILL_BACKGROUND_PROCESSES
- MANAGE_OWN_CALLS
- MODIFY_AUDIO_SETTINGS
- NFC
- READ_SYNC_SETTINGS
- READ_SYNC_STATS
- RECEIVE_BOOT_COMPLETED
- REORDER_TASKS
- REQUEST_COMPANION_RUN_IN_BACKGROUND
- REQUEST_COMPANION_USE_DATA_IN_BACKGROUND
- REQUEST_DELETE_PACKAGES
- REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
- SET_ALARM
- SET_WALLPAPER
- SET_WALLPAPER_HINTS
- TRANSMIT_IR
- USE_FINGERPRINT
- VIBRATE
- WAKE_LOCK
- WRITE_SYNC_SETTINGS

Figura 3 - Permissões Normais. Site: https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#normal-dangerous

A imagem anterior apresenta alguns recursos utilizados atualmente pelos principais aplicativos disponíveis no mercado. Recursos simples que fazem parte da nossa vida, como o alerta por vibração do aparelho, que é executado quando chega uma mensagem em nosso smartphone. Esse alerta utiliza o recurso "VIBRATE" do dispositivo, que não apresenta perigo e muito menos exponha a segurança do usuário do aparelho. Sendo assim classificada no grupo de "Permissões Normais".

Entre os principais recursos desse grupo, destacamos além do "VIBRATE", o uso do "BLUETOOTH", "INTERNET" e "NFC".

Permissões Perigosas

As permissões perigosas são aquelas que comprometem a privacidade do usuário e podem comprometer o sistema operacional e demais aplicativos. Para utilizar essas permissões, o aplicativo necessita alertar sobre a intenção da utilização, e durante a execução, o usuário necessita conceder a sua autorização expressa para a liberação da permissão, por meio de uma mensagem exibida em tela.

Um exemplo clássico, é encontrado nos programas de edição de fotos, onde o aplicativo solicita a permissão de acesso ao usuário as pastas de galeria de imagens do dispositivo.

Para conhecer as principais "Permissões Perigosas" verifique a **Figura 4**, retirada do site oficial do sistema Android.

Grupo de Permissão	Permissões
CALENDAR	• READ_CALENDAR • WRITE_CALENDAR
CALL_LOG	• READ_CALL_LOG • WRITE_CALL_LOG • PROCESS_OUTGOING_CALLS
CAMERA	• CAMERA
CONTACTS	• READ_CONTACTS • WRITE_CONTACTS • GET_ACCOUNTS
LOCATION	• ACCESS_FINE_LOCATION • ACCESS_COARSE_LOCATION
MICROPHONE	• RECORD_AUDIO
PHONE	• READ_PHONE_STATE • READ_PHONE_NUMBERS • CALL_PHONE • ANSWER_PHONE_CALLS • ADD_VOICEMAIL • USE_SIP
SENSORS	• BODY_SENSORS
SMS	• SEND_SMS • RECEIVE_SMS • READ_SMS • RECEIVE_WAP_PUSH • RECEIVE_MMS

Figura 4 - Permissões Perigosas. Site: https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#permission-groups

A **Figura 4** apresenta alguns recursos dos dispositivos que se utilizados de forma indevida, podem resultar em sérios problemas ao usuário e ao próprio aparelho. Imagine um aplicativo malicioso, que tenha acesso completo aos recursos de "SMS" do dispositivo. Esse aplicativo consegue enviar mensagens para números aleatórios ou até mesmo números da sua agenda, como se as mensagens fossem escritas por você!

Por esse e outros motivos o sistema operacional Android informa ao usuário, e solicita dele uma aprovação. Todos os recursos da lista de "Permissões Perigosas" são pertinentes, e destacamos o uso da localização por GPS, utilização de chamadas de voz, utilização de SMS, calendário, câmera e até mesmo acesso aos arquivos e contatos do dispositivo.

Uso de permissões de acesso aos recursos de Software

Os projetos executados nos sistemas operacionais inferiores ao Android 5.1, solicitavam a autorização do usuário para as permissões utilizadas no momento da instalação. A partir da versão 6.0, as permissões são autorizadas na primeira execução do aplicativo. A **Figura 5,** mostra um exemplo de solicitação da versão 5.1 e a **Figura 6,** mostra a solicitação das versões 6.0 ou superiores.

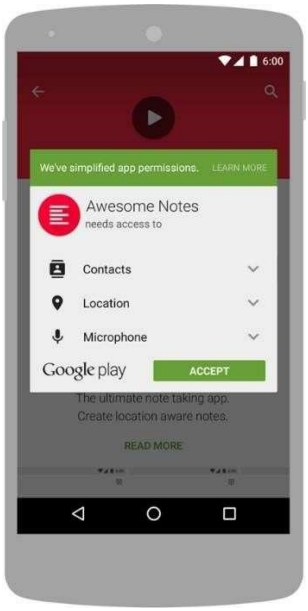


Figura 5 - Permissão nas versões 5.1 ou inferiores

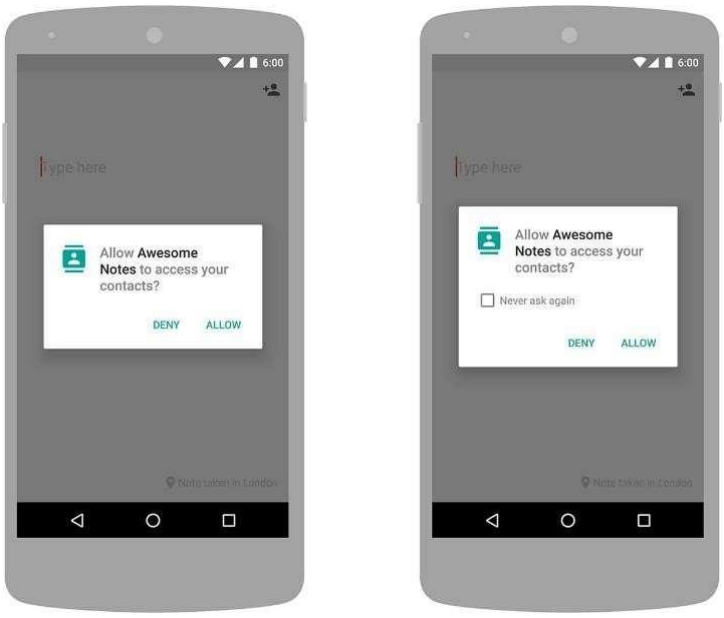


Figura 6 - Permissão nas versões 6.0 ou superiore

Uso de permissões de acesso aos recursos de Hardware

O sistema operacional Android também solicita permissões de uso de Hardware, e sabemos que nem todos os dispositivos são iguais, e muito menos oferecem todos os recursos como câmeras, GPS, Bluetooth e etc. Isso pode implicar negativamente em um projeto. Desta maneira, é importante o desenvolvedor declarar as permissões de maneira correta, evitando assimaborrecimentos do usuário. Imagine um jogo que utiliza o sistema de GPS e o usuário instala esse aplicativo em um dispositivo que não possui este recurso. Caso o desenvolvedor tenha esse cuidado com o código, podemos bloquear ofuncionamento deste jogo neste aparelho.

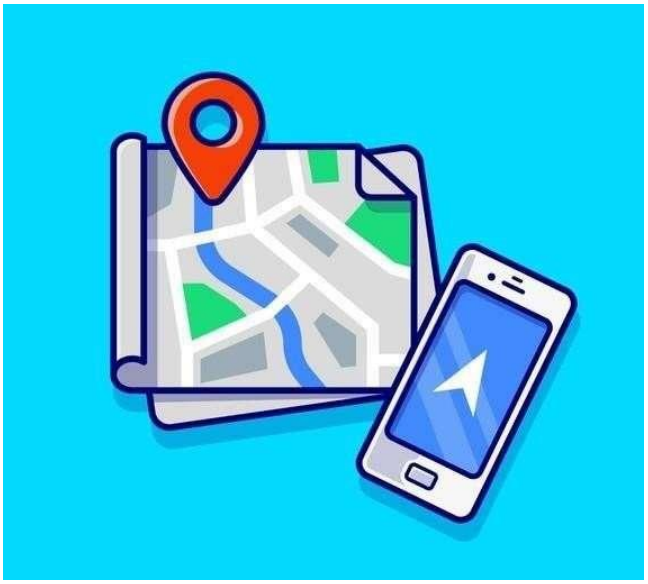


Figura 8 - Permissões GPS

O sistema operacional conta com outros tipos de permissão, desta forma é fundamental acompanhar na documentação oficial os outros tipos, e sempre que surgir alguma dúvida recorrer a esta documentação, disponível em: https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#normal-dangerous

Exemplo de utilização de permissões em um projeto

Este exemplo contempla apenas a solicitação de permissão ao sistema operacional Android, desta forma vamos criar um novo projeto para praticar como configurar o processo de permissões que altera o arquivo "**AndroidManifest.xml**", através da plataforma **Kodular** e realizar o teste com o dispositivo para exemplificar o processo de solicitação de permissão para o usuário.

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

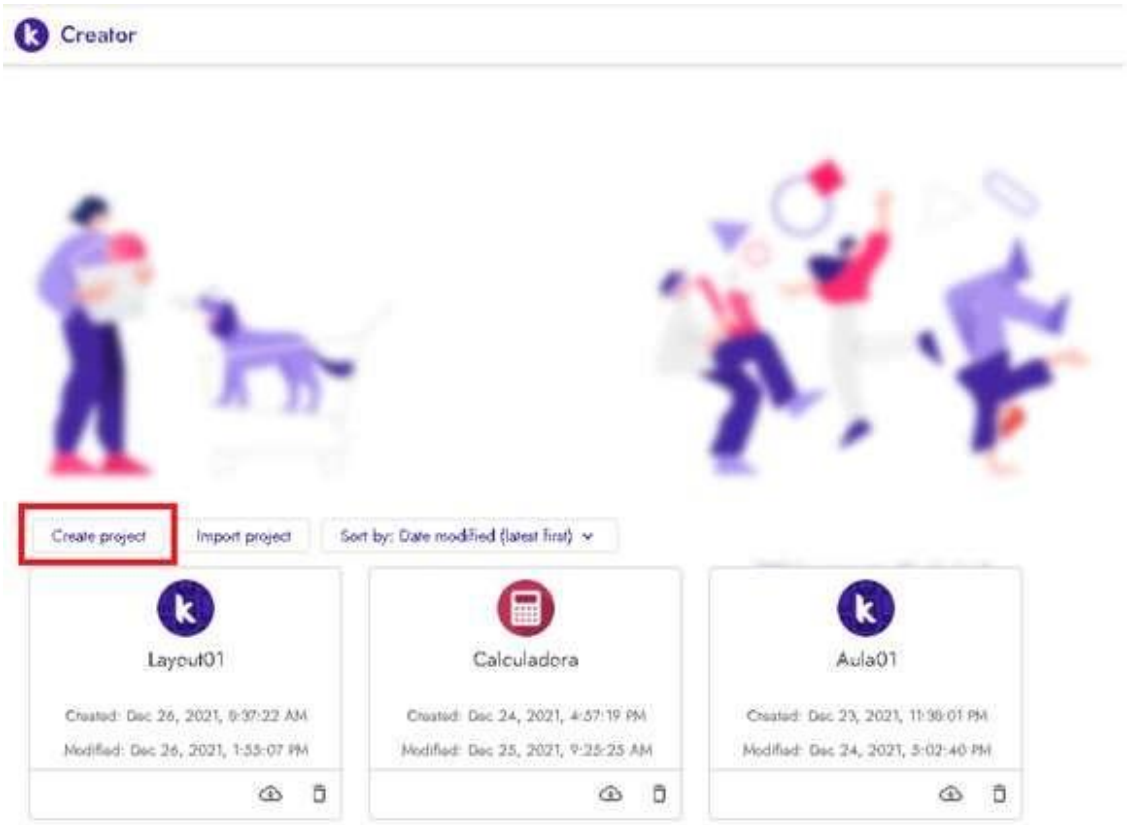


Figura 10 - Criando um novo projeto

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

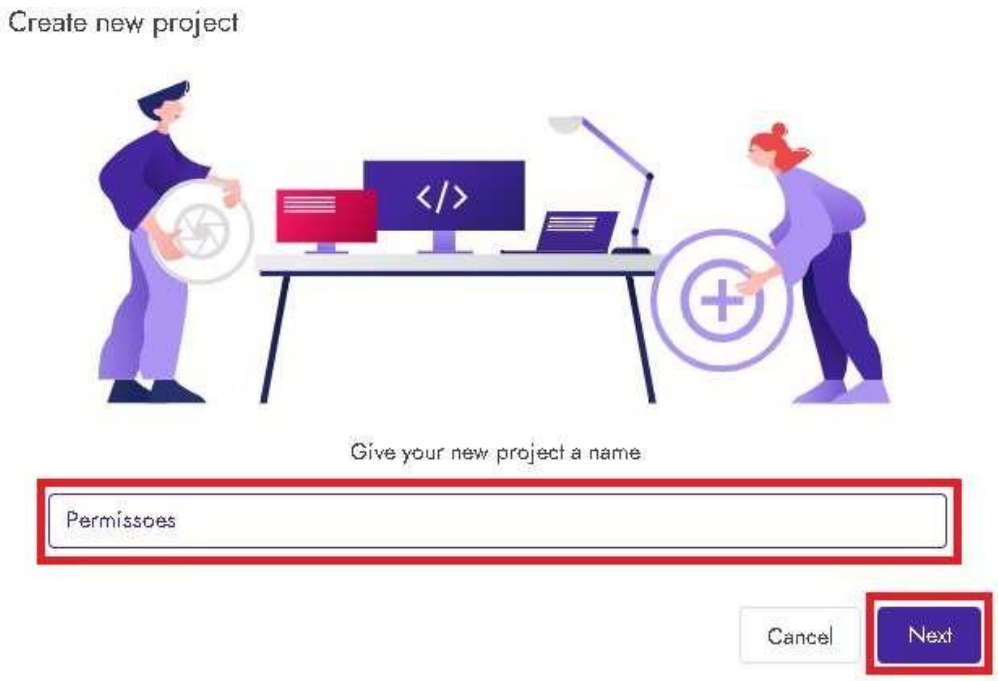


Figura 11 - Criando o projeto Compartilhar

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

Propriedade	Valor	Função
Align Horizontal	Center	Definir o alinhamento no sentido horizontal.
Align Vertical	Center	Definir o alinhamento no sentido horizontal.
Background Color	#000000FF	Definir a cor de fundo do aplicativo.
Title	Permissões	Definir o título da aplicação em desenvolvimento.

- Clique no componente **Label** na aba **Interface User** e insira na área de **VIEWER.**
- Altere as propriedades do objeto **Label.**

Propriedade	Valor	Função
Font Size	18	Definir o tamanho da fonte.
Font Bold	✓ Marcado	Definir a opção de negrito na fonte
Text		Definir o texto que será apresentado pelo **Label**.
Name	lbl_Permissao	Definir o nome do componente.

- Insira um componente **Button,** categoria **User Interface.**

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

Propriedade	Valor	Função
Width	200px	Definir a largura do componente button.
Font Bold	✓ Marcado	Definir a opção negrito
Font Size	18	Definir o tamanho da fonte
Text Alignment	Center	Define o alinhamento do texto em relação ao componente.
Text	Tirar Foto	Definir o conteudo a ser exibido do componente button.
Name	btn_Foto	Definir o nome do componente.

- Clique categoria **Media**.

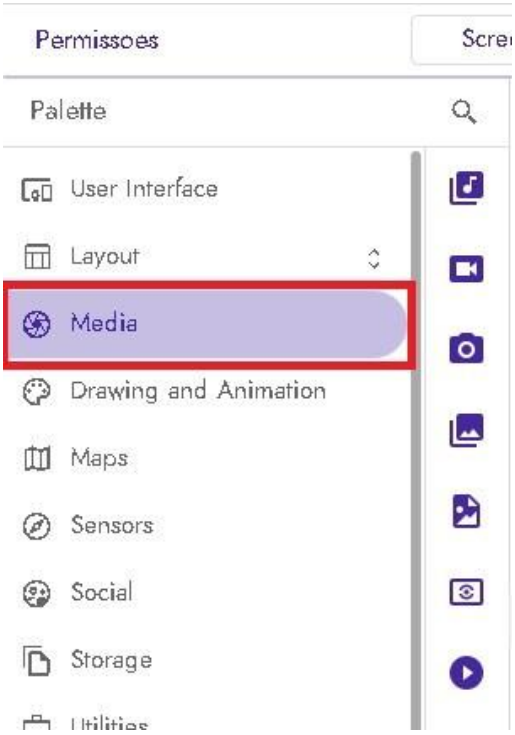


Figura 12 - Clique na categoria Media.

- Insira um componente Camera, categoria **Media**.

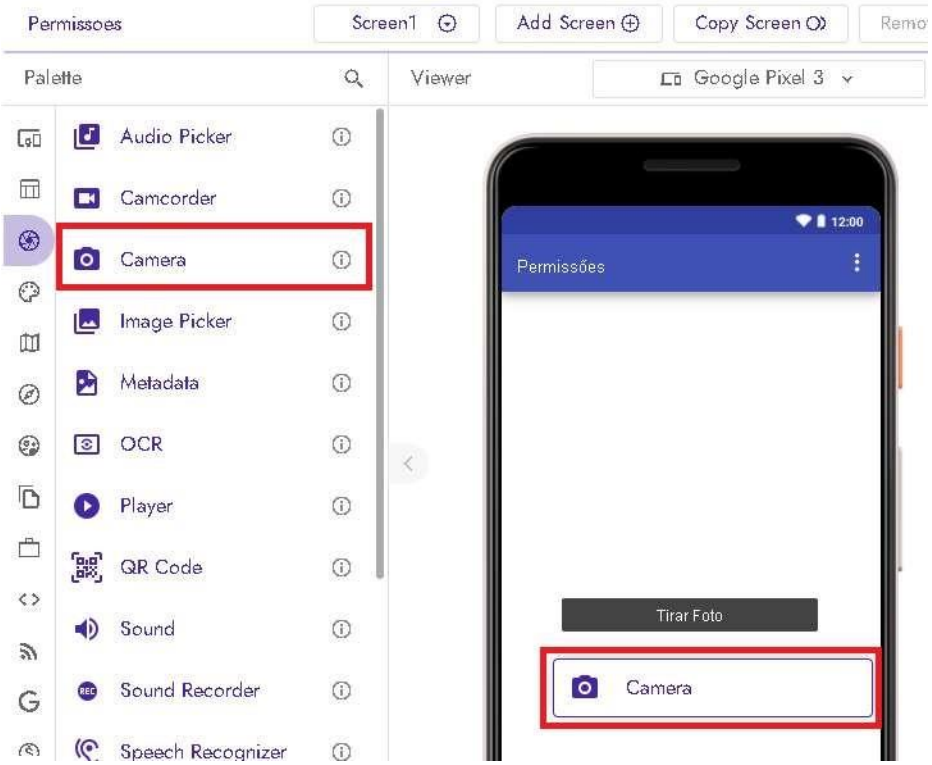


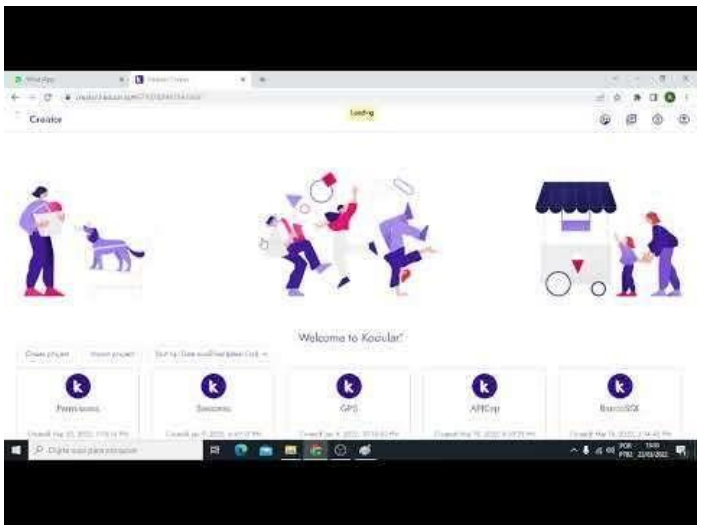
Figura 13 - Inserindo o componente Camera.

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

Propriedade	Valor	Função
Name	Cmr_Foto	Definir o nome do componente.

Finalizamos a construção do layout, mas caso tenha dúvidas sobre a inserção dos componentes e alterar suas propriedades, deverá assistir ao vídeo:

Agenda 05 – Criando a Interface Permissões, disponível em: https://youtu.be/8eueo0lkzr4



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

Figura 14 - Alterando a programação em blocos

- Clique no construtor **Screen**, clique e arraste o objeto **when Screen1.Initialize**

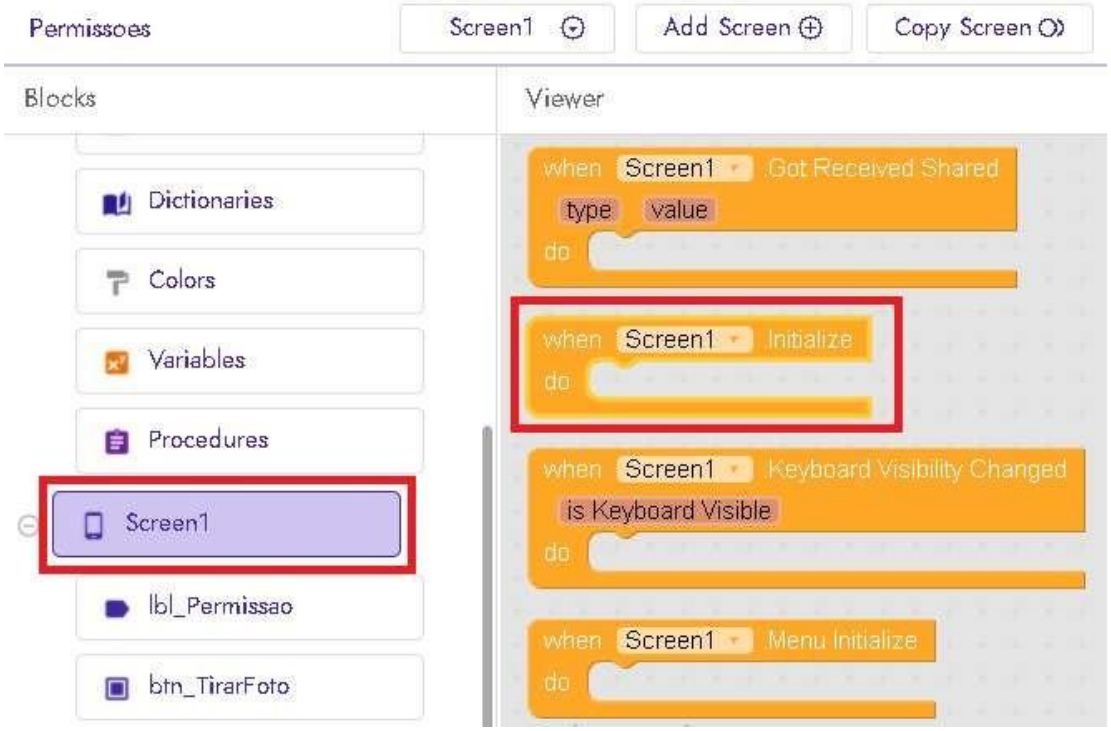


Figura 15 - Inserindo o objeto Screen.Initialize.

- Clique no construtor **Screen**, clique e arraste o objeto **call Screen1.Ask For Permission**

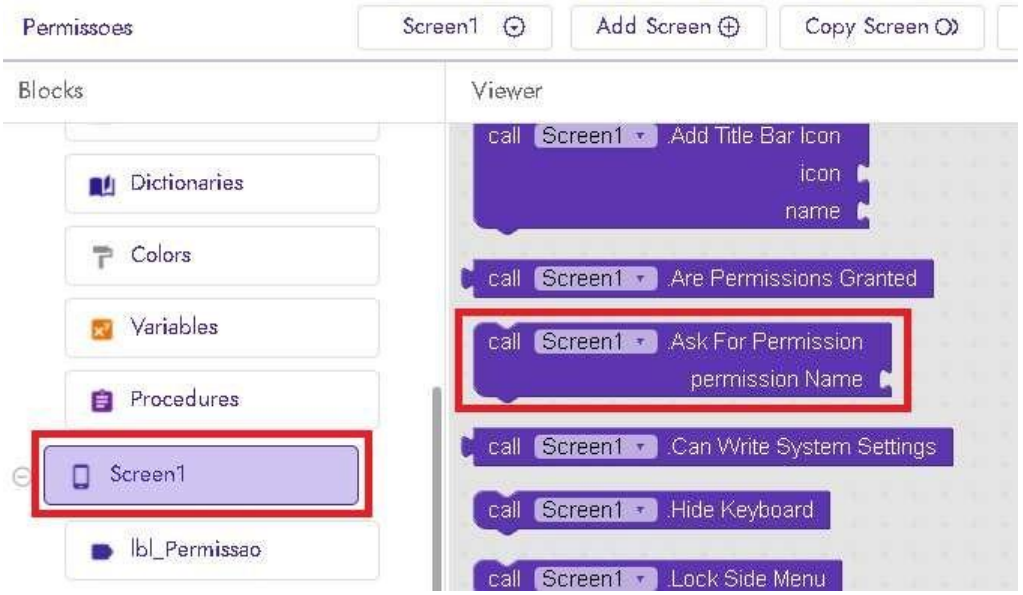


Figura 16 - Inserindo componente call screen1.Ask For Permission.

- Construa todos os códigos de programação em blocos, de acordo com a **Figura 18**.

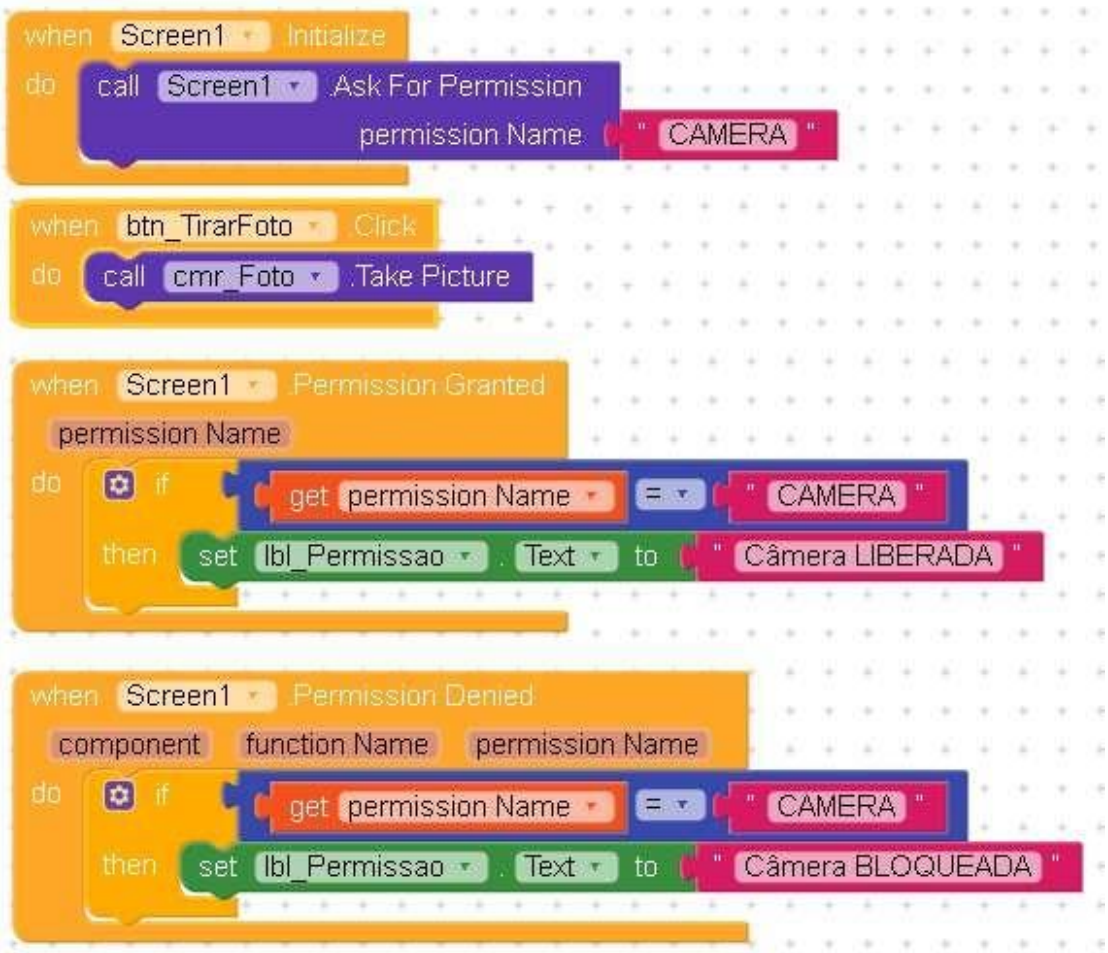
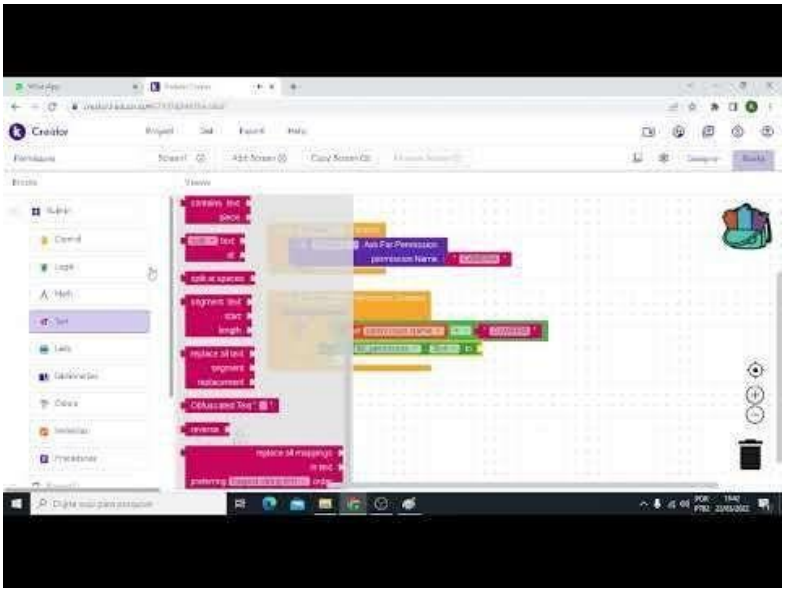


Figura 17 - Linguaguem de Programação em Blocos.

Caso tenha alguma dúvida em relação a criação do diagrama de blocos para a programação do aplicativo, assista ao vídeo:

Agenda 05 - Criando a programação em blocos Permissões, disponível em: https://youtu.be/Ui2DWwwgcRc



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

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

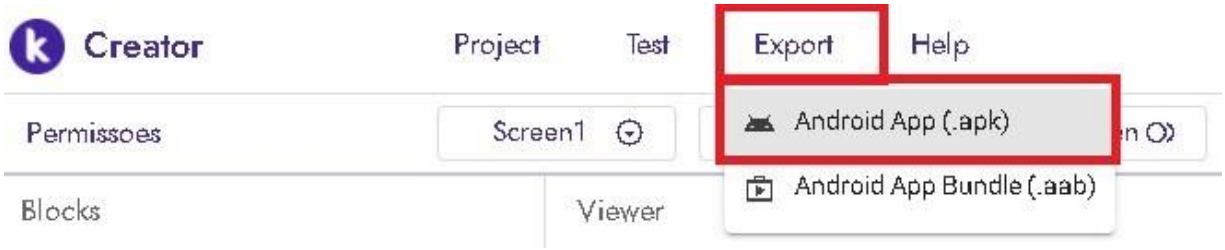


Figura 18 - Menu Export, opção Android App (.apk)

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

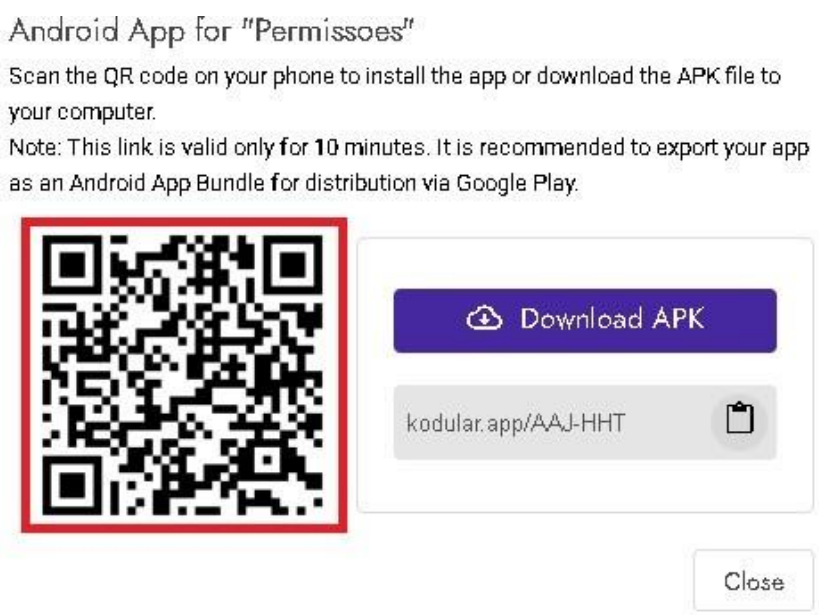


Figura 19 - Disponibilizando o Qrcode.

Ao iniciar o aplicativo, o usuário poder clicar em uma das opções para permitir ou não o acesso a Câmera do dispositivo móvel.

Figura 20 - Aplicativo com caixa de permissões.

Ao abrir o aplicativo, o código irá demonstrar se a permissão foi autorizada ou não. Através da mensagem exibida no componente **Label**.

Figura 21 - Aplicativo verificando a permissão.

Agora que já aprendemos sobre as permissões do sistema operacional Android, vamos praticar um pouco com a “Atividade Online”.