

NOTIFICAÇÕES

- mensagem que o Android exibe fora da IU do app para fornecer ao usuário lembretes, comunicados de outras pessoas ou outras informações oportunas do seu app.
- Os usuários podem tocar na notificação para abrir seu app ou executar uma ação diretamente da notificação.

Exibições em dispositivos

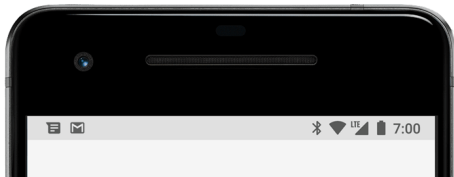


Figura 1. Os ícones de notificação aparecem no lado esquerdo da barra de status.

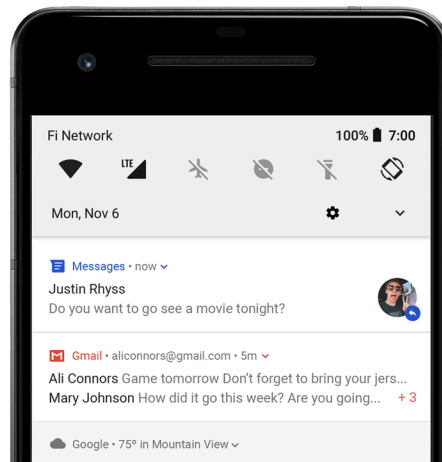


Figura 2. Notificações na gaveta.

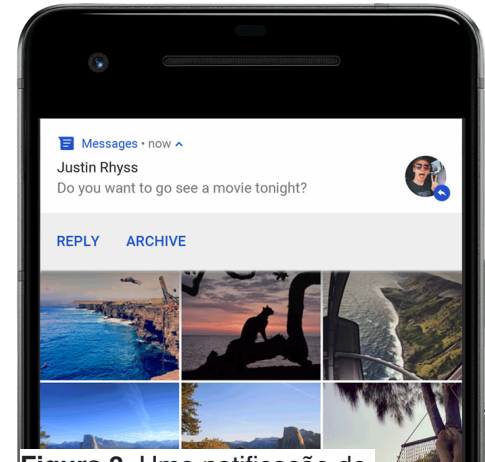


Figura 3. Uma notificação de informações básicas aparece na frente do app em primeiro plano.

NOTIFICAÇÕES

- mensagem que o Android exibe fora da IU do app para fornecer ao usuário lembretes, comunicados de outras pessoas ou outras informações oportunas do seu app.
- Os usuários podem tocar na notificação para abrir seu app ou executar uma ação diretamente da notificação.

Exibições em dispositivos

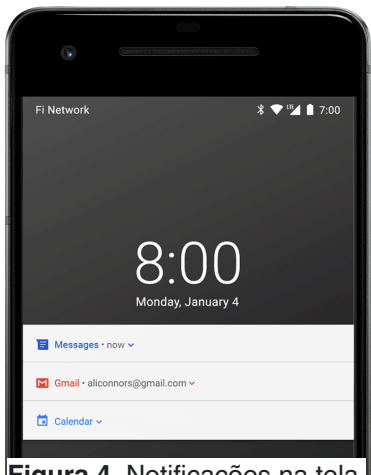


Figura 4. Notificações na tela de bloqueio com conteúdo sensível oculto.

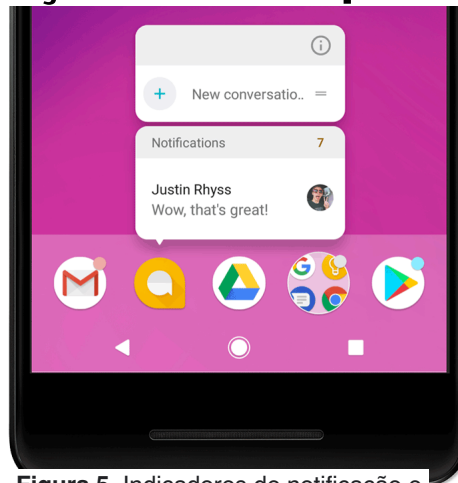


Figura 5. Indicadores de notificação e menu exibido após tocar e manter pressionado

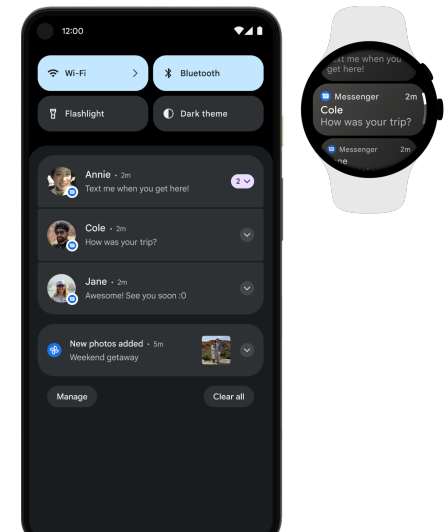
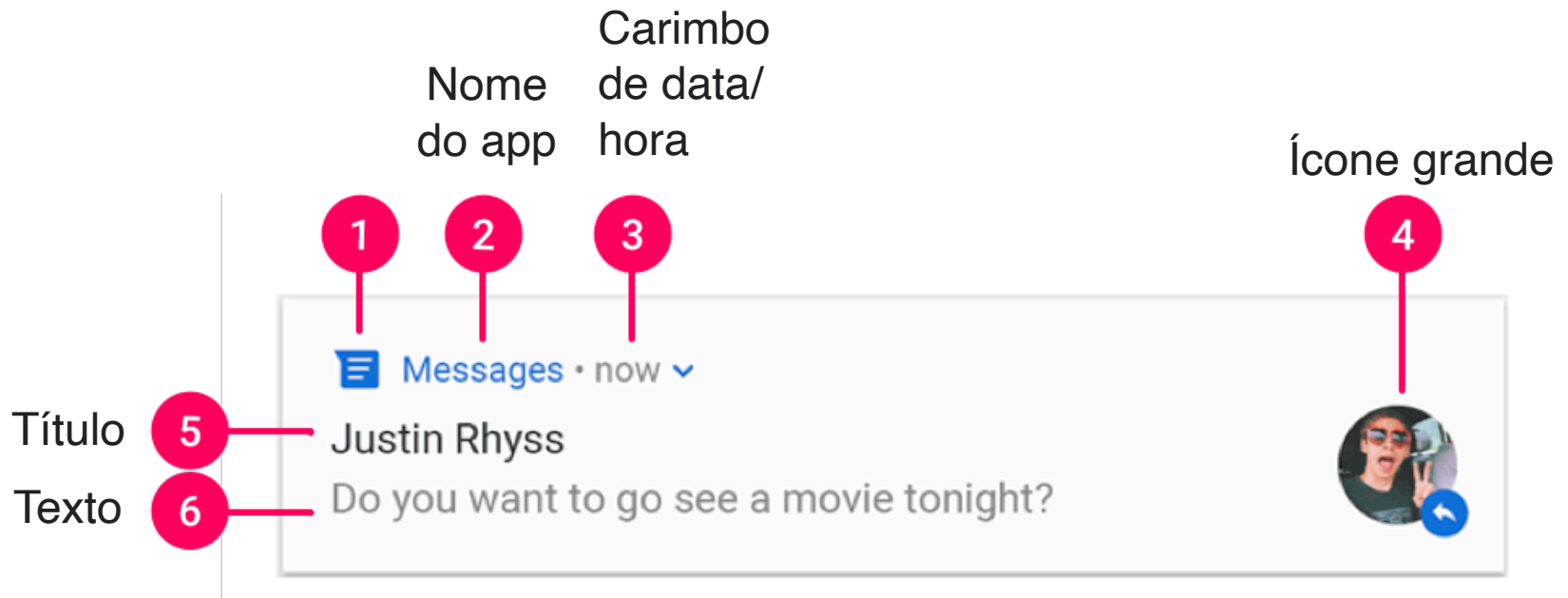


Figura 6. As notificações aparecem automaticamente em um dispositivo Wear OS pareado.

NOTIFICAÇÕES



EXPO NOTIFICATIONS

Notificações push Compatibilidade da plataforma

Dispositivo Android	Emulador Android	Dispositivo iOS	Simulador de iOS	Rede
✓	✗	✓	✗	✗

Notificações locais Compatibilidade da plataforma

Dispositivo Android	Emulador Android	Dispositivo iOS	Simulador de iOS	Rede
✓	✓	✓	✓	✗

EXPO NOTIFICATIONS

Notificações locais Compatibilidade da plataforma

Dispositivo Android	Emulador Android	Dispositivo iOS	Simulador de iOS	Rede
✓	✓	✓	✓	✗

As notificações locais são agendadas por um aplicativo localmente e entregues pelo mesmo dispositivo

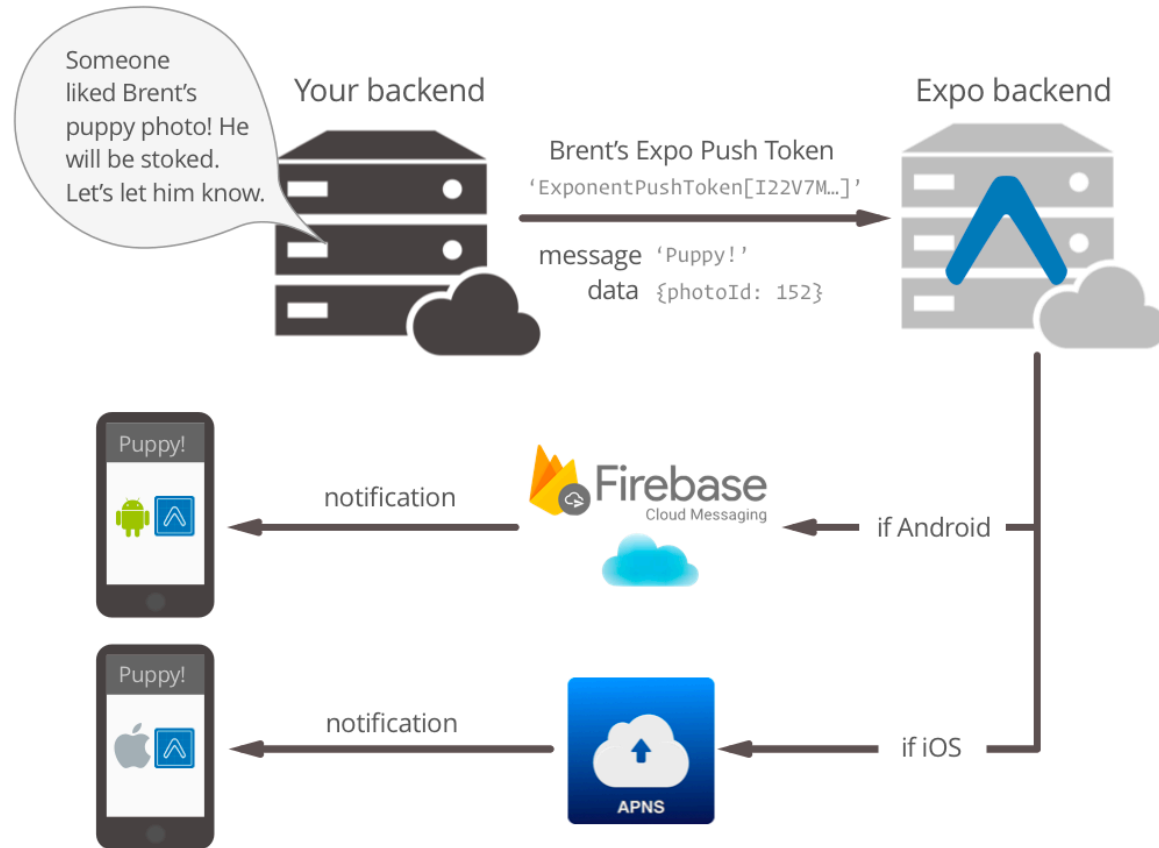
EXPO NOTIFICATIONS

Notificações push Compatibilidade da plataforma

Dispositivo Android	Emulador Android	Dispositivo iOS	Simulador de iOS	Rede
✓	✗	✓	✗	✗

As notificações push são enviadas por um servidor remoto (seu provedor) que envia essas notificações para os dispositivos nos quais o aplicativo está instalado.

EXPO NOTIFICATIONS



EXPO NOTIFICATIONS

Tratamento das notificações quando o aplicativo está em primeiro plano.

setNotificationHandler(handler)

Nome	Tipo	Descrição
manipulador	null <code>NotificationHandler</code>	Um único parâmetro que deve ser <code>null</code> (se você deseja limpar o manipulador) ou um <code>NotificationHandler</code> objeto.

Quando uma notificação é recebida enquanto o aplicativo está em execução, usando esta função você pode definir um callback que decidirá se a notificação deve ser exibida para o usuário ou não.

Quando uma notificação é recebida, `handleNotification` é chamado com a notificação recebida como argumento. A função deve responder com um objeto de comportamento em 3 segundos, caso contrário, a notificação será descartada. Se a notificação for tratada com sucesso, `handleSuccess` será chamada com o identificador da notificação, caso contrário (ou em timeout) `handleError` será chamada.

O comportamento padrão quando o manipulador não está definido ou não responde a tempo é não mostrar a notificação.

EXPO NOTIFICAÇÕES

Tratamento das notificações quando o

```
Notifications.setNotificationHandler({  
  handleNotification: async () => ({  
    shouldShowAlert: true,  
    shouldPlaySound: false,  
    shouldSetBadge: false,  
  }),  
});
```

o.

setNotificationHandler(handler)

Nome	Tipo	Descrição
manipulador	null <code>NotificationHandler</code>	Um único parâmetro que deve ser <code>null</code> (se você deseja limpar o manipulador) ou um <code>NotificationHandler</code> objeto.

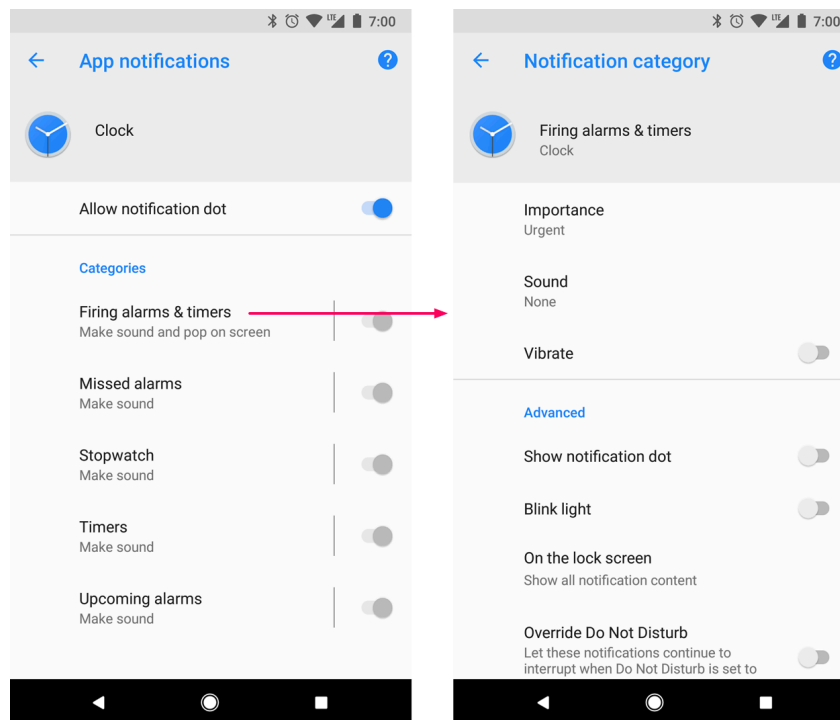
Quando uma notificação é recebida enquanto o aplicativo está em execução, usando esta função você pode definir um callback que decidirá se a notificação deve ser exibida para o usuário ou não.

Quando uma notificação é recebida, `handleNotification` é chamado com a notificação recebida como argumento. A função deve responder com um objeto de comportamento em 3 segundos, caso contrário, a notificação será descartada. Se a notificação for tratada com sucesso, `handleSuccess` será chamada com o identificador da notificação, caso contrário (ou em timeout) `handleError` será chamada.

O comportamento padrão quando o manipulador não está definido ou não responde a tempo é não mostrar a notificação.

CANAIS DE NOTIFICAÇÃO

- API de nível 26 (Android 8.0 e posteriores): todas as notificações precisam ser atribuídas a um canal.
- Para cada canal, você pode definir o comportamento visual e auditivo aplicado a todas as notificações. Usuário poderá definir configurações posteriores para cada canal.



EXPO NOTIFICATIONS

```
if (Platform.OS === 'android') {  
  await Notifications.setNotificationChannelAsync('default', {  
    name: 'default',  
    importance: Notifications.AndroidImportance.MAX,  
    vibrationPattern: [0, 250, 250, 250],  
    lightColor: '#FF231F7C',  
  });  
}
```

EXPO NOTIFICATIONS

requestPermissionsAsync(permissions)

Nome	Tipo	Descrição
permissões (opcional)	<code>NotificationPermissionsRequest</code>	Um objeto que representa a configuração para o escopo da solicitação.

Solicita ao usuário permissões de notificação de acordo com a solicitação. **Os padrões de solicitação solicitam ao usuário permissão para exibir alertas, definir a contagem de distintivos e reproduzir sons .**

Exemplo

Solicita ao usuário que permita que o aplicativo mostre alertas, reproduza sons, defina a contagem de distintivos e deixe a Siri ler mensagens por meio de AirPods.

```
import * as Notifications from 'expo-notifications';

export function requestPermissionsAsync() {
  return await Notifications.requestPermissionsAsync({
    ios: {
      allowAlert: true,
      allowBadge: true,
      allowSound: true,
      allowAnnouncements: true,
    },
  });
}
```

EXPO NOTIFICATIONS

```
const { status } = await Notifications.requestPermissionsAsync();
```

`requestPermissionsAsync(permissions)`

Nome	Tipo	Descrição
permissões (opcional)	<code>NotificationPermissionsRequest</code>	Um objeto que representa a configuração para o escopo da solicitação.

Solicita ao usuário permissões de notificação de acordo com a solicitação. **Os padrões de solicitação solicitam ao usuário permissão para exibir alertas, definir a contagem de distintivos e reproduzir sons .**

Exemplo

Solicita ao usuário que permita que o aplicativo mostre alertas, reproduza sons, defina a contagem de distintivos e deixe a Siri ler mensagens por meio de AirPods.

```
import * as Notifications from 'expo-notifications';

export function requestPermissionsAsync() {
  return await Notifications.requestPermissionsAsync({
    ios: {
      allowAlert: true,
      allowBadge: true,
      allowSound: true,
      allowAnnouncements: true,
    },
  });
}
```

EXPO NOTIFICATIONS

getPermissionsAsync()

Chamar esta função verifica as configurações de permissões atuais relacionadas a notificações. Ele permite que você verifique se o aplicativo tem permissão para exibir alertas, reproduzir sons, etc. Não há efeito de chamar isso para o usuário.

Exemplo

Verifique se o app tem permissão para enviar qualquer tipo de notificação (interrompendo e não interrompendo – provisório no iOS).

```
import * as Notifications from 'expo-notifications';

export async function allowsNotificationsAsync() {
  const settings = await Notifications.getPermissionsAsync();
  return (
    settings.granted || settings.ios?.status === Notifications.IosAuthorizationStatus.PROVISIONAL
  );
}
```

EXPO NOTIFICATIONS

```
const { status: existingStatus } = await Notifications.getPermissionsAsync();
```

getPermissionsAsync()

Chamar esta função verifica as configurações de permissões atuais relacionadas a notificações. Ele permite que você verifique se o aplicativo tem permissão para exibir alertas, reproduzir sons, etc. Não há efeito de chamar isso para o usuário.

Exemplo

Verifique se o app tem permissão para enviar qualquer tipo de notificação (interrompendo e não interrompendo – provisório no iOS).

```
import * as Notifications from 'expo-notifications';

export async function allowsNotificationsAsync() {
  const settings = await Notifications.getPermissionsAsync();
  return (
    settings.granted || settings.ios?.status === Notifications.IosAuthorizationStatus.PROVISIONAL
  );
}
```