

Usando GPS

MAC0448 - Programação para Redes de Computadores

Diogo Haruki Kykuta 6879613
Fernando Omar Aluani 6797226

1 Usabilidade

Ao abrir o aplicativo, o usuário irá ver a tela principal.

1.1 Tela Principal

A tela principal mostra de cara qual circular está selecionado (o padrão inicial é o Circular 1 (8012)), e um mapa centralizado na USP.

O mapa sempre mostra a rota do circular selecionado (em vermelho para o 8012, verde para o 8022) e a posição atual do usuário (um ponto ou seta azul).

Ao receber a posição do usuário, ele irá calcular qual o ponto na rota mais próximo ao usuário, e o próximo horário que esse circular está previsto para passar nesse ponto. Ao fazer isso, o seguinte acontece:

- O texto acima do mapa que fala qual circular está selecionado irá também mostrar o horário de chegada prevista do próximo ônibus;
- O mapa irá mostrar um marcador no ponto mais próximo. Ao clicar em tal marcador ele irá mostrar uma janelinha (como um balão) em cima do marcador, mostrando os dados que foram calculados: distância do terminal até esse ponto, tempo de chegada, horário de saída do terminal do ônibus e horário de chegada do ônibus nesse ponto.

O aplicativo sempre tentará pegar uma posição precisa do usuário, mas isso depende das capacidades e configurações do dispositivo. Os métodos existentes são:

- **Sinal de Celular:** pega a posição pela triangularização do sinal. É pouco precisa, mas é rápida, consome pouca energia e está quase sempre disponível;
- **Dados (3G/WiFi):** usa a internet para cálculo da posição. Normalmente é precisa, mas velocidade/consumo de energia depende da origem dos dados, e também pode haver outros gastos (com rede de dados 3G por exemplo).
- **GPS:** se o GPS estiver ligado, usa os satélites GPS para triangularização. É o método mais preciso, mas gasta mais energia, pode demorar um pouco inicialmente para conectar-se aos satélites, e o ambiente pode influenciar adversamente a conexão.

O mapa (Google Maps) necessita de conexão à internet para baixar os dados do mapa. No entanto, ele faz cache do mapa. Logo ele só irá necessitar internet de fato na primeira vez que o app é aberto, ou de tempos em tempos depois

se o cache for limpo. Como o mapa é aberto já centralizado na USP, é certeza que ele ira cachear as regiões importantes para o aplicativo, e uma vez que eles tenham sido carregados, dá pra ver o mapa da USP sem conexão à internet. O mapa também aceita gestos intuitivos com os dedos para movimentação do mapa, zoom, etc.

O menu de opções tem 4 opções:

- **Circular 1 (8012)**: muda o circular selecionado para este;
- **Circular 2 (8022)**: muda o circular selecionado para este;
- **Centralizar na USP**: move o mapa e altera o zoom para centralizar a câmera na USP;
- **Tabela de Horários**: abre a tela da tabela de horários.

1.2 Tela da Tabela de Horários

Essa tela mostra o circular e dia da semana selecionado, a duração (em minutos) da viagem de um ônibus (terminal -> volta na USP -> terminal) nesse dia, e a lista de horários de saída do terminal dos ônibus nesse dia.

O menu de opções possibilita a escolha do circular e a escolha do dia da semana.

Para voltar para a tela principal basta apertar o botão *Back* do seu dispositivo.

2 Estrutura do Código

O código escrito (não incluindo o gerado pelo ADT) foi dividido em 3 partes, relacionadas aos pacotes. Fora isso, também temos os recursos da aplicação.

2.1 Pacote principal (fefzjon.ep2.gps)

Aqui estão as classes principais do aplicativo:

- **MainActivity**: é a *activity* principal do aplicativo. A primeira a ser executada. Mostra o mapa com a rota do circular selecionado. Ao receber a localização do usuário, ele mostra no mapa o ponto na rota mais próximo ao usuário, e calcula os horários relacionados. O menu de opções permite selecionar qual circular é mostrado, centralizar o mapa na USP e mostrar as tabelas completas de horários de saída dos circulares no terminal Butantã.
- **TimeTableActivity**: *activity* secundária que mostra a duração da viagem e horários de saída do terminal para um dado circular em um dado dia. O menu de opções permite escolher o circular e o dia.
- **MarkerInfo**: classe que implementa uma interface do GoogleMaps para gerar o View mostrado ao usuário quando ele clica num marcador no mapa.

2.2 Pacote de Utilidade (fefzjon.ep2.gps.utilities)

Aqui estão as classes de utilidades usadas pelo aplicativo. Em sua maior parte são classes *singleton* que gerenciam os dados relacionados aos circulares (rotas e horários).

- **ConnectionHandler**: classe auxiliar que gerencia criação, uso e callbacks da API de Location do Android, a qual (entre outras coisas) envia à sua aplicação a posição do usuário. Ela sempre tenta adquirir uma posição precisa em intervalos curtos de tempo (5 segundos), mas isso pode mudar de acordo com os recursos do dispositivo.
- **RouteManager**: classe que trata o acesso e outras operações comuns para as rotas (sequência de pontos de latitude/longitude) dos circulares.
- **TimetableManager**: classe que trata o acesso a e outras operações comuns para os horários (duração e horários de saída) dos circulares.

2.3 Pacote de Diálogos (fefzjon.ep2.gps.dialog)

Aqui estão as classes de diálogos usadas pelo projeto. Todas estendem a *DialogFragment* e implementam um dialogo simples para escolha de 1 item dentre uma lista de possibilidades.

- **SelectBUSPDIALOGFragment**: dialogo para seleção do circular, de acordo com o código do ônibus.
- **SelectDayTypeDialogFragment**: dialogo para seleção do dia da semana (dia útil, sábado ou domingo).

2.4 Recursos

Fora os recursos padrão do Android, como o *strings*, *layout's* e *menu's*, só temos um recurso notável a mais: o **circulares.xml**, em *res/values*.

Esse recurso contém todos os dados dos circulares: latitudes e longitudes dos pontos da rota, duração da viagem e horários de saída para cada circular em cada dia da semana.

Todos esses dados foram pegos da tabela disponibilizada pelo monitor, que os adquiriu do site da SPTrans. A única diferença é que pontos iguais seguidos que existiam na lista de pontos original foram reduzidos a um único ponto.