

**Colégio Técnico de Campinas, Unicamp - UNICAMP**

**Felipe Melchior de Britto RA:18200  
Gabrielle da Silva Barbosa RA:18183**

## **Relatório Projeto de Estrutura de Dados II**

**Ferrovias na Europa**



**Campinas, 2019**

## OBJETIVOS DO PROJETO

A proposta era fazer um projeto em Android que fosse capaz de ler um arquivo de cidades e de caminhos entre cidades e, a partir deles, através de grafos e hashBuckets, mostrar ao usuário o caminho mais vantajoso entre duas cidades por ele escolhidas (de acordo com a distância ou o tempo), e também desenhá-lo no mapa ilustrativo. Além disso, também seria possível adicionar cidades e caminhos, que seriam salvos nos respectivos arquivos quando o programa fosse finalizado.

## DESENVOLVIMENTO

12/11:

- Optamos por fazer o projeto na linguagem C#, portanto criamos um projeto Android em branco no Visual Studio, na plataforma denominada "Xamarin".
- Majoritariamente trabalhamos com o design da página principal, adicionando Spinners que mostrariam as opções de cidades origem e destino, Buttons de busca, adicionar cidade e adicionar caminho, ImageView para o mapa, TextView para os resultados.

13/11

- Começamos a codificar, começando pela leitura de arquivos, usamos a classe Cidade do projeto "CidadesMarte" do semestre anterior com pequenas alterações e criamos uma classe InformacoesPercurso para guardar o tempo e as distâncias entre as adjacências, que guardamos numa matriz de adjacências.
- Tivemos alguns problemas com a leitura, pois em Xamarin não é exatamente a mesma forma quanto em C# tradicional, com algumas pesquisas em fóruns na internet conseguimos achar soluções para o problema.

14/11

- Colocamos as classes ListaSimples, HashBucket, Grafo, entre outras, que seriam necessárias para o funcionamento esperado, como foram feitas com outros propósitos, precisamos ajustá-las para o nosso projeto, alterações estas que não eram extremamente complexas e assim não comprometeram o propósito original da classe.
- Lemos as cidades e guardamos em ListaSimples<Cidade>
- Lemos os caminhos e guardamos em uma matriz de InformacoesPercurso indexada dos ids da cidade origem e da cidade destino

18/11

- Percebemos que não estávamos guardando as cidades no HashBucket e nem os caminhos no Grafo de Dijkstra, então começamos a adaptar essas classes para utilizar ListaSimples e variáveis do tipo Cidade e InformacoesPercurso.

- Também fizemos aparecer as opções de cidades origem e destino nos respectivos spinners através de uma `ArrayList<string>` com os nomes de cidades setada como adapter do tipo `SimpleListItem1`.

24/11

- Começamos a ter alguns problemas referentes ao próprio Visual Studio, pois alguns botões simplesmente sumiram, estando no código de layout, mas não aparecendo efetivamente no design da página.
- Depois de muitos “cleans” e “rebuilds” e até de fechar o Visual Studio, conseguimos resolver esse problema. Aparentemente, nem sempre ao se criar algo relacionado à página (como um button) o programa interno do visual studio salva a existência dessa declaração, atribuindo um identificador e uma posição para esse objeto, tivemos que algumas vezes realizar essa inserção manualmente.
- Também finalizamos o método de Dijkstra, utilizando apenas a distância como parâmetro para a escolha de melhor caminho.

25/11

- Iniciamos a parte de mexer com canvas para o desenho dos caminhos no mapa, criamos uma classe chamada “MyView” que herdava de View e implementava uma sobrecarga do método `OnDraw(Canvas canvas)` para o desenho através do canvas.
- Descobrimos que o Xamarin era mais diferente do que pensávamos do Android Studio (além de haver significamente menos informações úteis na internet) e, depois de sucessivas falhas, resolvemos começar a página de adição de cidades e deixar essa parte para mais tarde.

01/12

- Fizemos majoritariamente mudanças relacionadas aos desenhos de cidades e caminhos na imagem, devido aos problemas em se desenhar em uma imagem, não fizemos muito progresso.
- Criamos as páginas relacionadas à adicionar cidade e adicionar caminho e criamos também os botões que levariam para essas páginas.
- Surgiram problemas para fazer a conexão entre páginas.

02/12

- Corrigimos os problemas para conexão entre páginas através de pesquisas simples na internet, na qual pudemos encontrar diferenças entre Xamarin e Android Studio nos métodos utilizados para fazer conexão entre diferentes activities.
- Fizemos o código relacionado a página de criar uma nova cidade, para que a criação fosse realizada (mas ainda sem salvar no arquivo de cidades).
- Fizemos o código relacionado a criar um caminho entre duas cidades, na activity principal e na dedicada a propriamente criar um caminho.
- Fizemos as devidas verificações relacionados aos dois métodos acima.

- Além disso, colocamos radio buttons para que o usuário pudesse escolher se quisesse o melhor caminho em relação ao tempo ou à distância e adaptamos o método de Dijkstra para que verificasse isso também.

03/12

- Com muitos problemas para se desenhar na imagem, apagamos e recomeçamos a parte da realização do desenho. Conseguimos por fim desenhar na imagem da península ibérica os caminhos e cidades que estavam nos respectivos arquivos textos.
- A partir dos conhecimentos adquiridos, resolvemos também colocar um mapa em adicionar cidades, para que pudesse ser visualizada a coordenada digitada (toda vez que houvesse uma mudança nos textos de coordenada x e y ele passou a verificar se era possível desenhar no mapa, e, se sim, passou a desenhar um ponto vermelho no local).
- Encontramos um problema relacionado ao cálculo da distância e do tempo, durante a utilização alguns valores eram mudados (e não deveriam). Corrigimos fazendo uma simples mudança em um “for” que fazia com que o valor de distância de um determinado caminho fosse indevidamente alterado.
- O código para escrever no arquivo estava pronto, mas ainda não tinha sido testado, e quando testamos ocorreu um erro. Descobrimos, depois de pesquisas, que a utilização dos Assets só podia ser feita para leitura e, para gravação, deveríamos utilizar o armazenamento interno. Como solução, já que no armazenamento interno podia não haver o arquivo texto, quando vamos ler verificamos se existe no armazenamento interno e, se não, lemos as versões originais guardadas no Assets. Quando o usuário sai pela primeira vez, ele salva no armazenamento interno (automaticamente criando um novo arquivo se não existir) e a partir daí sempre será utilizado de lá.
- Fizemos, finalmente, com que as cidades e caminhos adicionados fossem salvos nos arquivos texto ao se fechar a aplicação ou terminar um processo específico (override do método OnStop() da activity).
- Comentamos todo o código apropriadamente.

## CONCLUSÕES

Uma aplicação útil para o aprendizado. Através das dificuldades encontradas em se utilizar uma nova plataforma pudemos ampliar nossos conhecimentos relacionados a diferentes linguagens e IDEs. A utilização dos algoritmos e métodos vistos em sala de aula ajudaram na compreensão de suas utilidades e funções, facilitando a retenção de tudo o que foi visto em sala de aula.