

DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA CÁLCULO DE MÉTODOS DE PROGRAMAÇÃO LINEAR NA PLATAFORMA ANDROID

João Paulo Barbosa Glória

Graduado em Ciências da Computação / UCAM
jpb.gloria@msn.com

Daniel Souza Caldas

Graduando em Ciências da Computação / UCAM
daniels.caldas@gmail.com

Thiago Muniz Barbosa

Doutorando em Engenharia e Ciências dos Materiais / UENF
thiagomuniz@gmail.com

RESUMO

Este trabalho apresenta um aplicativo, desenvolvido em Java e orientado a objetos, que pode ser instalado em variadas versões do Android, e utilizada no meio acadêmico e também profissional. Através da interface, desenvolvida para o Sistema Operacional(SO), o usuário poderá acessar através de um smartphone ou tablet o aplicativo e realizar cálculos complexos de Programação Linear com rapidez e facilidade. O código foi escrito de modo que possa ser utilizado em outras plataformas, alterando apenas a interface, e incrementado com novas funcionalidades. O aplicativo permite um número ilimitado de restrições e variáveis para resolução dos problemas apresentados, e retorna como resultado os valores das iterações e a solução final dos problemas, permitindo ao usuário analisar cada resultado obtido.

Palavras-chave: *Programação Linear; Pesquisa Operacional; Android; Simplex.*

1. Introdução

Desenvolvida durante a Revolução Industrial, a Pesquisa Operacional se espalhou por diversas áreas e se expandiu com os anos e a necessidade. Seu auge de desenvolvimento ocorreu na Segunda Guerra Mundial, quando cientistas desenvolveram métodos para otimizar o uso dos recursos. A Programação Linear (PL) é uma de suas áreas, e seus métodos abordam o melhor uso de recursos, seja para maximização ou minimização.

Há no mercado, atualmente, softwares que realizam os cálculos necessários para resolver problemas pelos métodos de PL, porém os mais completos são restritos à microcomputadores. Além de restringir o uso aos computadores de mesa, os softwares não são intuitivos, dificultando o uso para novos usuários.

Com a popularização dos dispositivos que utilizam a plataforma Android, observou-se uma oportunidade de desenvolver um aplicativo para o SO que fosse capaz de solucionar de maneira rápida e eficiente os problemas apresentados através dos métodos Simplex, Simplex Duas Fases e Gráfico.

O Java foi a linguagem escolhida para o desenvolvimento do algoritmo pela sua versatilidade e recursos. Orientada a Objetos e facilmente adaptável a outras plataformas, deixa em aberto a possibilidade de aplicativos para outros Sistemas Operacionais.

1.1. Objetivo

. O objetivo deste trabalho é o desenvolvimento de um aplicativo para dispositivos móveis capaz de solucionar problemas matemáticos através dos métodos Simplex e Gráfico de PL.

2. Pesquisa Operacional

2.1. Histórico

Os primeiros métodos de PO surgiram junto com a Revolução Industrial, mas foi durante a Segunda Guerra Mundial, próximo aos momentos finais, que a PO foi criada. Os exércitos precisavam organizar o seu poderio de forma a ter um melhor aproveitamento de cada arma, e que cada tropa recebesse a quantidade necessária de equipamento e munição para os combates. Era necessário que um equipamento, que estava em quantidade mais limitada, fosse posicionado de forma a ser movida, com um menor custo e mais rapidamente possível para outro ponto em que fosse necessária a sua utilização. Não só os recursos bélicos, mas também suprimentos estavam cada vez mais escassos. Água, comida e medicamentos eram limitados e era necessário garantir a correta distribuição dos mesmos (EHRlich, 1976).

A logística tornou-se um problema a ser solucionado em meio à guerra. Para resolvê-lo, cientistas americanos e britânicos trabalharam juntos no desenvolvimento de métodos matemáticos, que deram origem à área hoje conhecida como PO. O nome foi escolhido devido a seu desenvolvimento para fins de soluções operacionais, e seu desenvolvimento teve como objetivo primário solucionar a alocação de recursos bélicos e suprimentos (EHRlich, 1976).

Com o sucesso da aplicação da PO na solução de controle de recursos durante a Segunda Guerra Mundial, O método foi exportado para outras áreas que não a militar após o término da guerra. Grandes indústrias, comércios e governos passaram a utilizar a PO para aperfeiçoar a produção e alocação de recursos, não mais bélicos, mas produtos em estoque, transporte de mercadorias, para reabastecimento de lojas ou mesmo frete, e recursos financeiros, mão-de-obra, policiamento, entre outros (EHRlich, 1976).

2.2. Programação Linear

Segundo Duckworth (1972), PO é uma técnica adotada em situações onde existem vários produtos a serem fabricados, com o auxílio de máquinas e a necessidade de programas com o intuito de decidir acerca da máquina a ser utilizada para a fabricação de cada produto, levando em conta a produção máxima, o custo mínimo ou qualquer que seja o critério de eficácia. A PL surgiu como um dos mais importantes ramos da programação matemática devido a sua vasta aplicação prática. Com o avanço tecnológico ocorrido a partir da segunda metade do século XX, os algoritmos se tornaram ainda mais eficientes e favoráveis para a resolução de uma larga variedade de problemas envolvendo questões de decisão em vários domínios, entre os quais podem ser citados o planejamento da distribuição e produção de produtos, no planejamento de curto prazo em aproveitamento hidroelétricos e nas decisões ligadas às políticas microeconômicas e macroeconômicas de governo dos países (SOUSA, 2014)

A origem do problema de otimizar uma função linear sujeita às restrições lineares decorreu dos estudos de Fourier sobre os sistemas lineares de inequações em 1826. Apesar disso, Kantorovich, em 1939, foi quem fez notar a importância da prática destes problemas, tendo ele criado um algoritmo capaz de soluçona-los. (SOUSA, 2014)

2.2.1. Modelagem

Em PO, a modelagem consiste em traduzir a realidade empírica para sentenças lógicas e dados objetivos. Através dessa tradução é possível estabelecer um modelo matemático. Esse é primeiro passo para a resolução do problema de PO, e é denominado formulação. Nele é decidido, por julgamento humano, os aspectos do sistema real que deverão ser incorporados ao modelo e os que poderão ser ignorados, as suposições que poderão ser consideradas e as que podem ser descartadas. O processo porém está sujeito a erros e falhas de comunicação. Para o estabelecimento do modelo de um problema, não existem técnicas precisas, capazes de permitir o estabelecimento do modelo de um problema. (TAHA, 2008)

O segundo passo para a resolução do problema é a dedução do modelo, ou seja, sua análise e resolução através de algoritmos específicos. Essa solução, atenta aos métodos numéricos em computação, sugere uma tomada de decisão. Para a sua sustentação, é necessário recorrer ao terceiro passo, que é a interpretação de uma solução do sistema real. Se não for validado, o modelo deve ser reformulado, e assim por diante. Esse processo é chamado de modelagem. (TAHA, 2008)

2.2.2. Método Gráfico

É possível encontrar a solução ótima de um problema graficamente, porém apenas quando envolver duas variáveis.

Para resolver o problema graficamente, o primeiro passo é estabelecer os dois eixos que irão representar as quantidades de x e y . O próximo passo é encontrar o conjunto de soluções viáveis do problema. Para tal, pode ser utilizada a representação gráfica imposta por cada uma das restrições, ou seja, determinar qual subárea do espaço e seria aceita por cada restrição. Algumas dessas restrições são de fácil interpretação (LACHTERMACHER, 2007)

2.2.3. Método Simplex

A forma tabular do método simplex é a mais conveniente de se utilizar para resolver um problema de PL. No lugar de dicionários, um quadro Simplex (matricial) é usado para registrar apenas as informações essenciais, como os coeficientes das variáveis, as constantes das restrições e as variáveis básicas e as não básicas. (LACHTERMACHER, 2007)

É necessário, portanto, simplificar a forma de um dicionário, estabelecendo um quadro equivalente. Depois devemos verificar como cada operação analítica realizada pode ser automatizada através de regras de comando. Por fim, devemos verificar como tomar a decisão de parada do algoritmo. (LACHTERMACHER, 2007)

As variáveis originais do problema são as não básicas e as variáveis de folga são as básicas (lado esquerdo das equações). O próximo passo para a obtenção do quadro inicial é a modificação do dicionário inicial para se obter o dicionário inicial modificado, que servirá como ponto de partida para a formação do quadro Simplex inicial. Para essa modificação é necessário trocar de lado todas as variáveis do problema (neste rol estão incluídas as variáveis originais, as de folga e Z) (LACHTERMACHER, 2007)

No dicionário inicial modificado as variáveis da equação que representam as funções objetivo trocaram de dados, isto é, quando queríamos aumentar o valor de Z procurávamos as variáveis

da equação que tinham coeficientes positivos. Como as variáveis mudaram de lado na equação, agora é preciso procurar as com sinal negativo. A decisão de parar ocorre quando não tivermos mais variáveis com sinais negativos, ou seja, quando todos os coeficientes forem positivos ou iguais à zero (LACHTERMACHER, 2007).

3. Metodologia

No desenvolvimento da aplicação, foi utilizada a programação iterativa e incremental como modelo de desenvolvimento de software. Esse modelo permitiu que fosse feita a análise de requisitos, análise e designer do projeto, desenvolvimento, teste e integração de cada pequena iteração definida, reduzindo a taxa de erros e garantindo a definição e implementação corretas do escopo de iteração. Ao final de cada iteração temos um incremento no software que é consolidada com uma nova versão do sistema.

Na primeira etapa do desenvolvimento do projeto, foi iniciada a programação do núcleo do sistema que vai realizar todos os cálculos necessários para obtenção do resultado desejado. Mediante a necessidade de uma linguagem de programação orientada a objetos, o Java foi adotado para a implementação do algoritmo. Para a resolução de um problema, o usuário possuía um conjunto de funções que era dividido em função objetivo e restrições. Foi necessário a criação de um modelo do tipo Função para poder representar esses dados no sistema, e então desenvolvido a interface *Funcao*, a classe abstrata *FuncaoAbstrata*, que tem por objetivo armazenar os valores das funções, o operador, que pode ser \leq , \geq ou $=$, além de métodos para calcular e inserir nas funções os valores das variáveis de folga e variáveis artificiais.

Após a realização dos primeiros testes, foi desenvolvida a interface para utilização através do Android, e realizados testes, visando obter resultados quanto ao funcionamento do algoritmo desenvolvido no S.O. móvel.

Para a criação da interface no android precisamos da junção de um arquivo XML e de uma Activity, que é uma classe do java que representa uma tela. O arquivo XML é utilizado quando o layout é definido antes da compilação do programa, como foi utilizado na tela inicial do aplicativo, onde é solicitado ao usuário que informe a quantidade de variáveis e restrições. a classe controladora Principal é responsável inicialmente por gerar todos os campos necessários do problema para que possa ser preenchido pelo usuário. Ao final, ele deverá clicar no botão resolver para dar continuidade a resolução e, nesse momento, o controlador faz todas as verificações necessárias, como se todos os campos obrigatórios fossem informados e se não houvesse nenhum erro sistêmico. Caso esteja tudo correto ela cria uma nova thread e nela também é feita a identificação se todos os dados informados pelo usuário se trata de um problema Uma fase ou Duas fases. Ela também tem que identificar se o usuário optou pela Maximização ou Minimização do problema, além de identificar se os valores das variáveis podem ser positivos ou negativos, após toda essa etapa ela se encarrega de instanciar o Simplex uma fase ou Simplex duas fases.

Após a execução desses passos é chamada a classe controladora Resultado, que tem como layout o arquivo resultado.xml. Neste caso o layout é todo definido antes da compilação do sistema. Há abas para exibição do resultado final e de todas as iterações do problema, além de componentes do tipo Label vazios para exibição dos resultados assim que terminada sua execução. Assim essa controladora é responsável por identificar se existe algum erro durante a execução do Simplex, caso tenha erro, ou a solução seja impossível de se calcular, ela apresenta uma mensagem de erro, e caso não tenha erro, ela preenche a tela com os resultados.

Com o sucesso da primeira etapa, foi iniciado o desenvolvimento do método simplex Duas Fases. Para esse, foi desenvolvido um código capaz de tratar o problema matemático

apresentado até obter a resolução do problema, ou as condições necessárias para ser aplicado o método anterior, que será chamado ao término da execução do algoritmo de Duas Fases.

A classe `SimplexDuasFases` é responsável por executar as suas ações específicas. É também uma classe de inicialização do método Simplex Duas Fases, e também faz a verificação da condição de parada deste tipo de simplex, caso todas as condições de parada sejam atendidas ela passa a chamar a classe `SimplexUmaFase` para dar continuidade na execução do problema. Para se obter uma instancia dessas classes, é necessário passar dois argumentos no construtor da classe a ser chamada, que são: o número de variáveis e o número de restrições do problema a ser resolvido.

Na última etapa do desenvolvimento do aplicativo, foi programado todo o cálculo necessário e plotagem das retas do método gráfico na tela do dispositivo móvel.

No aplicativo, é possível apenas realizar o cálculo pelo método gráfico para funções com duas variáveis, sendo essa verificação realizada pela classe controladora Principal.

4. Resultados

Ao final de todas as etapas de desenvolvimento, temos então um aplicativo denominado “Simples PL”, representado nas telas abaixo com suas respectivas descrições.

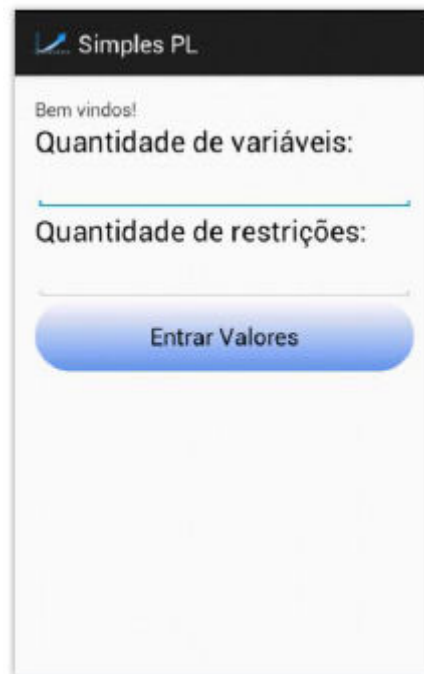
Na Figura 1 temos a área de trabalho, de um dispositivo móvel com o sistema operacional Android, onde temos instalado um atalho para acesso ao Simples PL.



Figura 1– Tela com atalho para acesso ao Simples PL.

Fonte: Própria

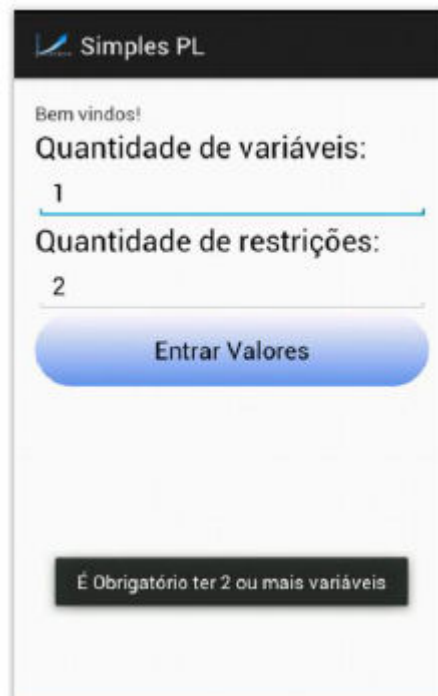
Como vemos na Figura 2, temos a tela inicial do Simples PL, uma tela simples e direta, sem poluição visual, para uma melhor usabilidade do aplicativo. Nela temos a possibilidade de informar apenas a quantidade de variáveis e de restrições que serão utilizadas no problema escolhido pelo usuário, e a seguir deverá ser clicado no botão “Entrar Valores”.



A tela principal do aplicativo 'Simples PL' apresenta o seguinte layout: no topo, um cabeçalho preto com o ícone de uma seta verde e o texto 'Simples PL' em branco; abaixo, o texto 'Bem vindos!' em cinza; em seguida, o rótulo 'Quantidade de variáveis:' seguido por um campo de entrada com uma barra de limite superior; depois, o rótulo 'Quantidade de restrições:' seguido por um campo de entrada com uma barra de limite superior; e, no centro, um botão azul arredondado com o texto 'Entrar Valores'.

Figura 2– Tela principal do aplicativo.
Fonte: Própria

Nos campos de entrada de variáveis e restrições, somente é permitido a digitação de números, porém duas restrições do sistema impedem que o usuário informe valores menores que 2 para variáveis ou restrições, sendo a mensagem informada conforme Figura 3.



A tela do aplicativo 'Simples PL' mostra o mesmo layout da Figura 2, mas com o valor '1' digitado no campo 'Quantidade de variáveis' e o valor '2' no campo 'Quantidade de restrições'. O botão 'Entrar Valores' permanece visível. Na base da tela, uma caixa de mensagem preta com o texto 'É Obrigatório ter 2 ou mais variáveis' indica o erro.

Figura 3– Tela com mensagem de erro no campo de variável.
Fonte: Própria

Ao entrar valores satisfatórios no sistema, temos a tela de processamento de preparação dos campos que serão preenchidos posteriormente, conforme Figura 4. O tempo de processamento dessa tela será de acordo com o número de variáveis e restrições informadas. Quanto maior o número informado nesses campos, maior será o tempo de processamento. Esse tempo, porém, pode variar de alguns milésimos de segundos até alguns poucos segundos, isso dependerá diretamente do dispositivo utilizado no momento, qual o processador e quantidade de memória disponível para processamento.

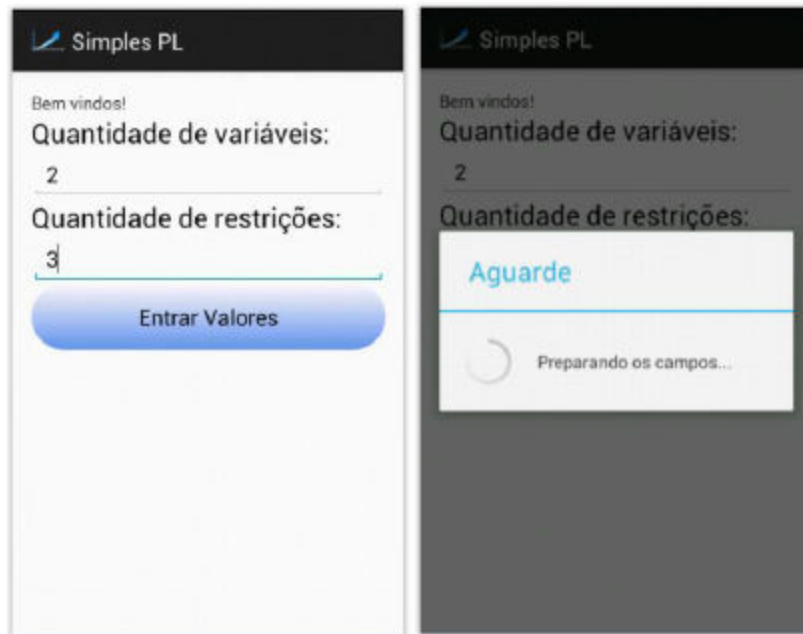


Figura 4– Tela de entrada com valores satisfatórios e de processamento dos campos.

Fonte: Própria

Temos então, na Figura 5, a tela gerada de acordo com a quantidade de variáveis e restrições informadas pelo usuário. O sistema já traz como padrão a opção “Max” selecionada, para maximização do problema, porém pode-se alterar para minimização trocando para a opção “Min”.

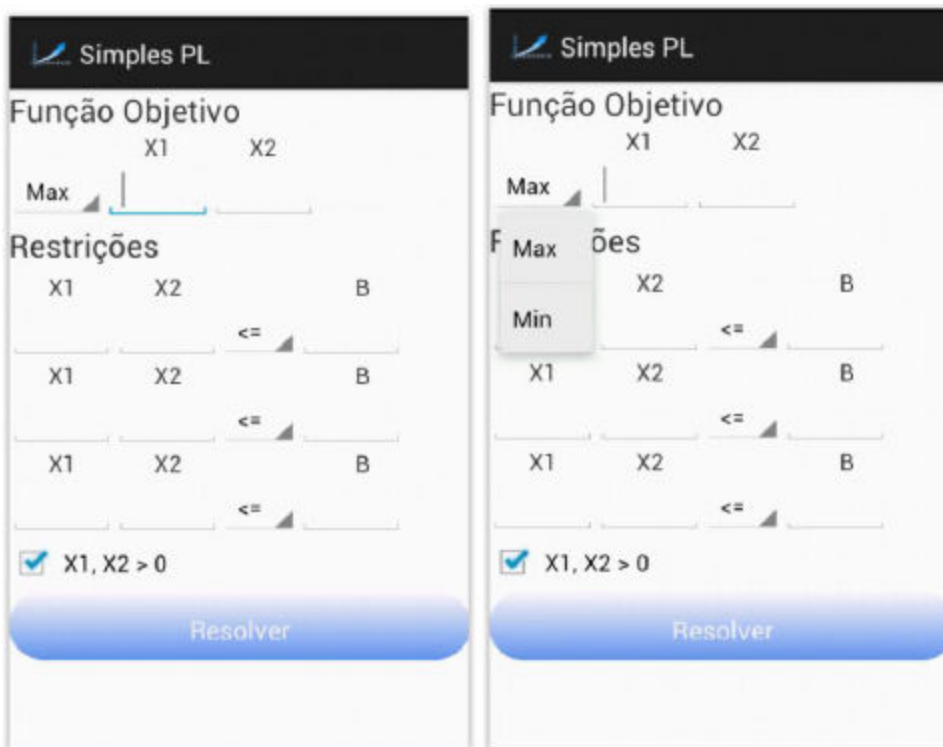


Figura 5 Tela de entrada de valores e de escolha para minimização ou maximização.
Fonte: Própria

Caso o usuário clique no botão “Resolver” antes de preencher todos os campos, tanto para a função objetivo, quanto para as restrições, é informada uma mensagem de erro, solicitando o preenchimento dos campos, conforme Figura 6.

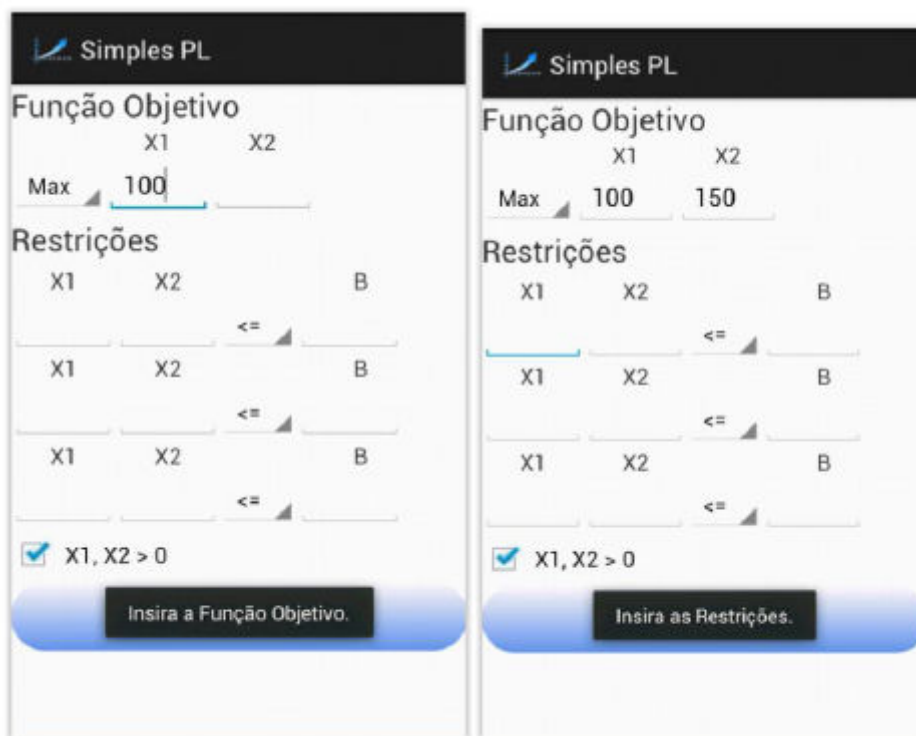


Figura 6– Tela com mensagem de erro na função objetivo e com mensagem de erro nas restrições.
Fonte: Própria

O usuário precisa escolher qual o sinal que vai definir aquela função. Ele poderá escolher, \leq , \geq ou $=$, sendo que a opção \leq já vem marcada como padrão. É nesta etapa que o aplicativo irá definir a utilização de qual método Simplex irá utilizar, se será o Simplex Uma Fase, ou Simplex Duas Fases.

Ao preencher todos os campos corretamente, o usuário deverá clicar no botão “Resolver”, e receberá um aviso para aguardar o processamento do resultado final pelo programa, conforme Figura 7.



Figura 7– Tela com valores preenchidos corretamente.

Fonte: Própria

A próxima etapa é a exibição da tela de resultados, onde a mesma possui três abas, conforme pode ser visto na Figura 24. A primeira aba, que já vem selecionada, encontra-se a função objetivo e as restrições informadas na tela anterior, e abaixo toda a solução do problema. Para acessar as demais abas, o usuário poderá deslizar o dedo na tela do smartphone, caso o seu dispositivo tenha uma tela de touchscreen, ou poderá acessar as opções “Iterações” ou “Gráfico”. Podemos visualizar passo a passo todos os dados das iterações até o resultado final, conforme Figura 8.



Figura 8– Tela com resultado do problema e onde mostra todas as iterações do problema.
 Fonte: Própria

Na Figura 9 temos a resolução do problema com os dados informados pelo usuário a partir do método gráfico. Quando se é informado valores que podem ser resolvidos pelo método gráfico, esta aba ficará disponível para exibição.

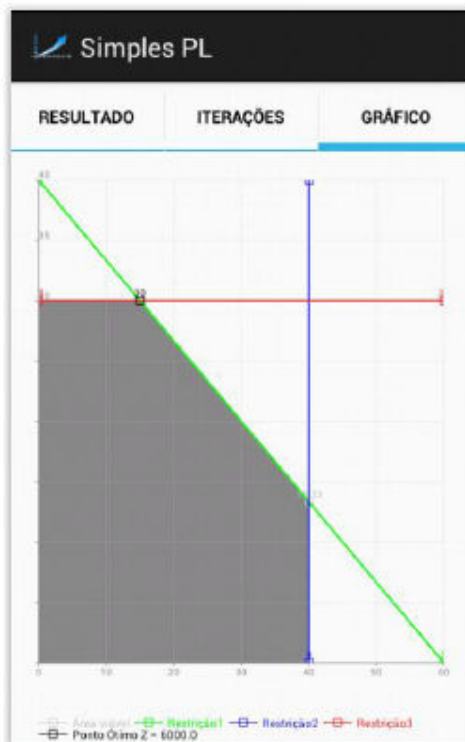


Figura 9– Tela com exibição do método gráfico.
 Fonte: Própria

Existem alguns problemas que não podem ser resolvidos pelo método Simplex, e outros que, por algumas combinações de valores, não são válidas na definição de um resultado, como na Figura 10. Uma tela mostra que não há uma solução possível de calcular com os valores informados.

The figure consists of two side-by-side screenshots of a mobile application interface for solving Linear Programming (LP) problems using the Simplex method.

Left Screenshot (Input Screen): Titled "Simplex PL". It displays the objective function: "Função Objetivo" with coefficients 100 for X_1 and 150 for X_2 , and a "Max" indicator. Below are three constraints, each with coefficients for X_1 and X_2 , a comparison operator (\leq), and a right-hand side value (B):
1. $2X_1 + 3X_2 \leq 120$
2. $1X_1 + 0X_2 \leq -40$
3. $0X_1 + 1X_2 \leq 30$
At the bottom, there is a checkbox for "X1, X2 > 0" which is checked, and a blue "Resolver" button.

Right Screenshot (Result Screen): Titled "Simplex PL" with a sub-header "RESULTADO". It shows the objective function: "Max +100X1 +150X2". Below it, the constraints are listed: "Sa +2X1 +3X2 <=120", "+1X1 <=-40", and "+1X2 <=30". A large blue rounded rectangle at the bottom contains the text: "Solução impossível de calcular."

Figura 10– Tela de entrada de valores e de exibição de solução impossível de calcular.

Fonte: Própria

5. Conclusão

Foram desenvolvidos algoritmos, usando a linguagem de programação Java, para resolver os cálculos e iterações relacionadas aos métodos Simplex, Simplex-Duas fases e Gráfico da área de PL. Com os algoritmos desenvolvidos, elaborou-se uma interface para permitir sua utilização por usuários do sistema Android. Após a conclusão da interface e integração do código a ela, iniciaram-se os testes, com o intuito de verificar se o aplicativo era funcional, de fácil manuseio e capaz de solucionar problemas mais complexos.

Em testes realizados com outros softwares para desktop, o aplicativo desenvolvido mostrou o mesmo resultado de valor de saída.

Referências

ANDROID(Ed.). **AndroidDeveloper Tools.** Disponível em: <http://developer.android.com/tools/help/adt.html>>. Acesso em: 28 fev. 2015.

"ANDROID" (Ed.). **KitKat 4.4:** Smart, simple, and truly yours. 2014. Disponível em: <http://www.android.com/intl/pt-BR/versions/kit-kat-4-4/>>. Acesso em: 15 nov. 2014

DUCKWORTH, Eric. Guia à pesquisa operacional. Tradução de Leonidas H. B. egemberg, Octanny Silveira Mota. São Paulo: Atlas, 1972. 159p., il.

EHRlich, Pierre Jacques. Pesquisa operacional: curso introdutório. São Paulo: Atlas, 1976. 189p., il.

LACHTERMACHER, Gerson. Pesquisa operacional na tomada de decisões: modelagem em Excel. 3.ed. rev. e atual. Rio de Janeiro: Elsevier/ Campus, 2007. 213p., il.

SOUSA, Jorge Alberto Mendes de. **SIG2005.** Disponível em: <http://pwp.net.ipl.pt/deea.isel/jsousa/Doc/SIG2005.pdf>>. Acesso em: 08 dez. 2014.

TAHA, Hamdy A. Pesquisa operacional. Tradução de Arlete Simille Marques. 8.ed. São Paulo: Pearson Prentice Hall, 2008. 359p., il. ISBN 978-85-7605-150-3.

THIAGO BARROS. Techmundo (Ed.). Cinco anos de Android: relembre a história e todas as versões do sistema. 2013. Disponível em: <http://www.techtudo.com.br/noticias/noticia/2013/09/cinco-anos-de-android-relembre-historia-e-todas-versoes-do-sistema.html>>. Acesso em: 23 set. 2013.