

# **Go: Sistema de alocação de voluntários para a Cidadão Pró-Mundo**

**Michel Carvalho  
Vinícius Aguiar**

---

## SUMÁRIO

---

1	PROBLEMA	3
2	MODELO CONCEITUAL	4
2.1	Preparação dos dados	4
2.2	Formulação	4
2.2.1	Algoritmo genético	4
2.2.2	Função objetivo	5
3	DOCUMENTO DE REQUISITOS	6
3.1	Modelo	6
3.2	Instalação	6
3.3	Uso	6
4	RESULTADOS	8

---

## PROBLEMA

---

A proposta apresentada pela ONG Cidadão Pró-Mundo consiste na alocação de seus voluntários em turmas pré-estabelecidas, respeitando na medida do possível as diversas preferências e restrições, impostas respectivamente pelos professores e entidade.

Inicialmente foram fornecidos dados referentes a unidade de Campinas, de 170 voluntários e 22 turmas. Nestes estão inclusos os cadastros dos intrutores com seu nível de inglês, comprometimento, data de ingresso, turma anterior, função, e preferências de período, turma e faixa etária. Foram descritos os tipos, períodos e progressões das turmas; como também os voluntários que devem seguir a mesma escala. Listamos então as preferências por ordem de prioridade decrescente:

- Balancear o comprometimento das turmas;
- Equilibrar o número de voluntários alocados por turma;
- Manter o nível de inglês dos voluntários superior ao de seus alunos;
- Respeitar as preferências de período, turma e faixa etária;
- Conciliar voluntários que devem ser alocados na mesma escala;
- Considerar a proporção de veteranos e calouros, homens e mulheres.

É importante notar que pela natureza das restrições algumas são mais facilmente satisfeitas, como a primeira acima.

---

## MODELO CONCEITUAL

---

Foi escolhida a linguagem de programação R e o software Rstudio, dada a familiaridade dos membros com o software e além disso por ser um software livre. Foi identificado que uma forma para resolver o problema seria modelá-lo como uma variação do clássico problema de designação, procuramos técnicas que pudessem resolver este problema de programação inteira. Diante disso nos deparamos com um problema semelhante, também de designação, chamado "Nurse Scheduling Problem" e sua resolução pela metodologia de algoritmos genéticos. Devido a isso, utilizamos a abordagem de otimização de algoritmos genéticos com uma função objetivo penalizada, como detalhado posteriormente.

### 2.1 PREPARAÇÃO DOS DADOS

Com o objetivo de diminuir o processamento desnecessário foram realizadas alterações nas entradas. Foram filtrados os EAs do Dia e Owners, para não constarem na alocação. Devido a isso foram eliminados os dados repetidos na tabela de voluntários com mesma escala. A partir disso, foram comparados os níveis de inglês de turmas de adultos e adolescentes. Por fim foram calculadas as constantes de uso geral e preferências.

### 2.2 FORMULAÇÃO

#### 2.2.1 *Algoritmo genético*

Pode ser descrito como uma metodologia de busca de soluções para problemas de otimização, inspirada em técnicas de biologia evolutiva, como seleção natural e recombinação. Suncitamente um algoritmo genético segue um ciclo evolutivo em que após gerar uma população inicial, ocorre uma seleção natural e recombinação dos indivíduos que tem maior adaptabilidade, isto é, tem uma maior função objetivo (também chamado fitness neste contexto). Logo ocorrem mutações aleatórias com pequena probabilidade

na nova população, que mantém uma boa variedade de indivíduos. Este processo é repetido até chegarmos a uma adaptabilidade aceitável para o problema, ou caso nosso poder computacional ou tempo se esgotem. Neste projeto foi utilizada a biblioteca Genetic Algorithm (GA), que implementa algumas funcionalidades da heurística. Em particular as funções seleção e passagem de gerações foram preservadas. Alternativamente foram alterados os processos de criação de população inicial, recombinação e mutação. Estes últimos foram implementados para preservar a característica do problema de alocação, o uso de números inteiros.

### 2.2.2 Função objetivo

É utilizada em problemas de otimização para medir a qualidade de uma solução. Neste caso ela é composta inicialmente pelas preferências dos voluntários por períodos, acompanhamento de turma ou nível, e faixa etária. É complementada então pelas restrições, que contribuem negativamente para seu valor, magnitude baseada na ordem de importância das mesmas. Ao inserir estas últimas variáveis na função objetivo dizemos que a mesma está penalizada. Atualmente a função objetivo incrementa em 1 para cada preferência satisfeita, e decrementa em valor absoluto pelas diferenças entre os valores obtidos de médias de turma e os valores esperados. Por exemplo, o valor esperado de voluntários por turma é 7, e a turma 5 tem 3 voluntários. Logo ao analisar essa turma fitness decrementa em 4. Foi escolhida essa formulação simples por facilidade computacional, no entanto ainda faltam mais análises de qualidade da solução.

---

## DOCUMENTO DE REQUISITOS

---

### 3.1 MODELO

O cliente espera que gestemos uma solução com os seguintes requisitos:

- Requisito 1: É necessário satisfazer o maior número de preferências possíveis, tendo como a principal o período.
- Requisito 2: O uso de software livre é imprescindível.
- Requisito 3: É esperada que saída seja em formato CSV, com os campos Turma e Voluntário.

### 3.2 INSTALAÇÃO

Go é um aplicativo desenvolvido na linguagem de programação R, que faz uso de bibliotecas para processamento de sua heurística e desenvolvimento de interfaces. Para sua utilização no sistema operacional Windows é necessário fazer download e instalar sequencialmente seu interpretador [R](#) e seu ambiente de desenvolvimento [RStudio](#), ambos amplamente disponíveis.

Seguimos então para instalação de suas bibliotecas, disponível através da execução do código `libraries`. Para isto deve-se selecionar o código inteiro e apertar em Run.

### 3.3 USO

O usuário pode então abrir o código do app go pelo aplicativo RStudio e executá-lo ao apertar o botão Run App.

Logo é possível fazer o upload das entradas do tipo CSV em padrões pré-estabelecidos e selecionar as preferências desejadas.

O app possui dois botões de rolagem que servem para aumentar as medidas de população e geração, que na forma de otimização por algoritmos genéticos podem ser

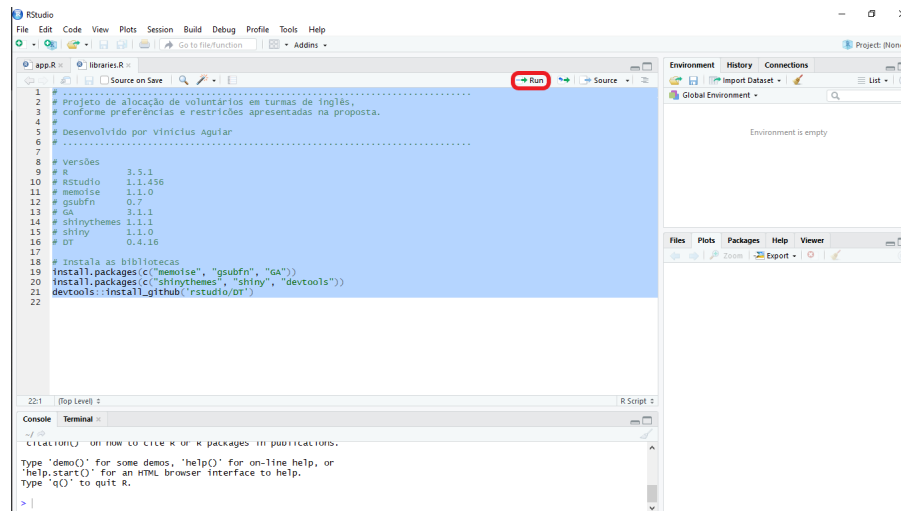


Figura 1: Como instalar as bibliotecas

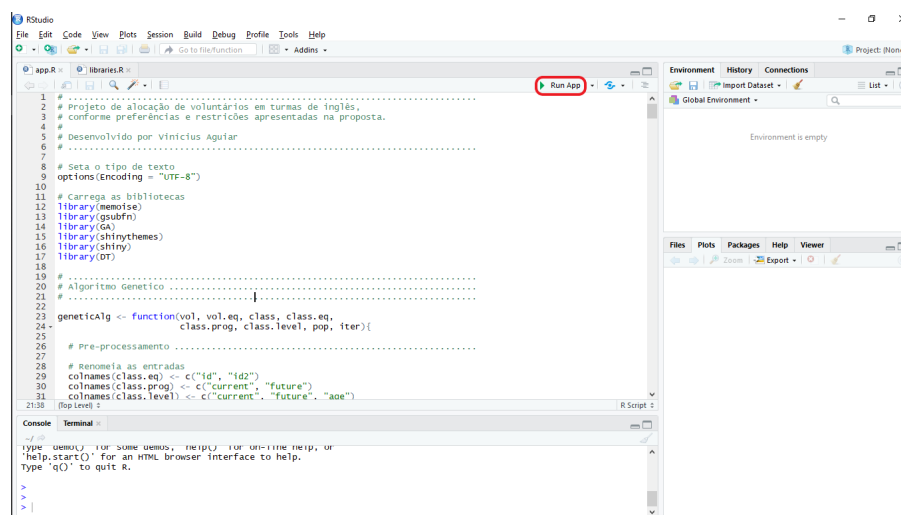


Figura 2: Como rodar o aplicativo

utilizadas para aumentar a qualidade da solução. Entretanto valores elevados destas variáveis aumentam o tempo de processamento.

É possível observar também a existência dos botões "Otimizar" e "Gerar CSV", que respectivamente geram a solução e após a geração da mesma realizam o download do arquivo de saída em CSV.

Para interromper o programa é necessário apenas fechar a aba do navegador utilizado. Note que neste ponto o software RStudio gera um aviso que a sessão foi terminada.

## RESULTADOS

Visualmente os resultados podem ser facilmente analisados e as turmas podem ser alteradas antes de se gerar a saída. Através desta heurística são geradas diferentes configurações finais com qualidades semelhantes em tempos razoáveis de processamento.

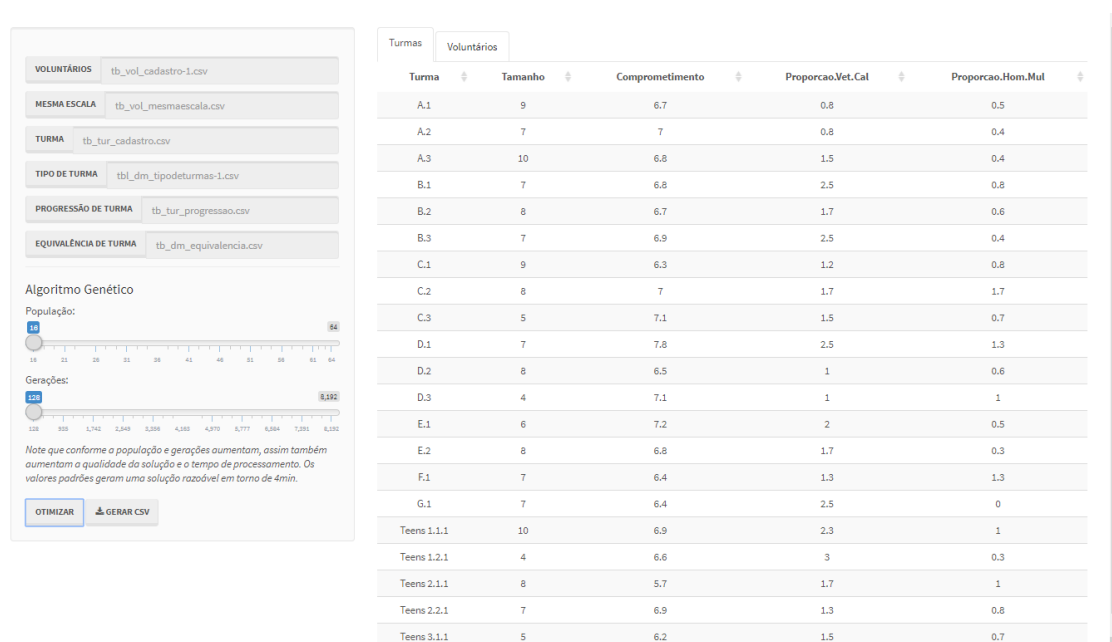


Figura 3: Exemplo de interface do aplicativo

No entanto por se tratar de uma aproximação de uma solução ótima não há garantia que consigamos chegar exatamente na melhor solução possível. O trabalho de encontrar as melhores opções para que o algoritmo tenha boa convergência também é manual, um contratempo ainda não solucionado.

Esperamos em próximos trabalhos encapsular o aplicativo de forma que o usuário não necessite fazer downloads de softwares terceiros, mesmo livres. Ou até hospedarmos gratuitamente em [shinyapps.io](https://shinyapps.io), escolha possível dado o pouco uso do mesmo durante o mês.