

Guia de uso do Mooshak

1. Introdução

O Mooshak é um sistema baseado na Web para gerir concursos de programação. Ele avalia automaticamente os programas submetidos e fornece feedback instantâneo.

2. Objetivo

Neste contexto, vamos usar o Mooshak para auxiliar a validação dos exercícios de programação propostos nas aulas laboratoriais. Pretende-se, também, que a sua utilização possa ultrapassar os limites geográficos e temporais das sessões laboratoriais.

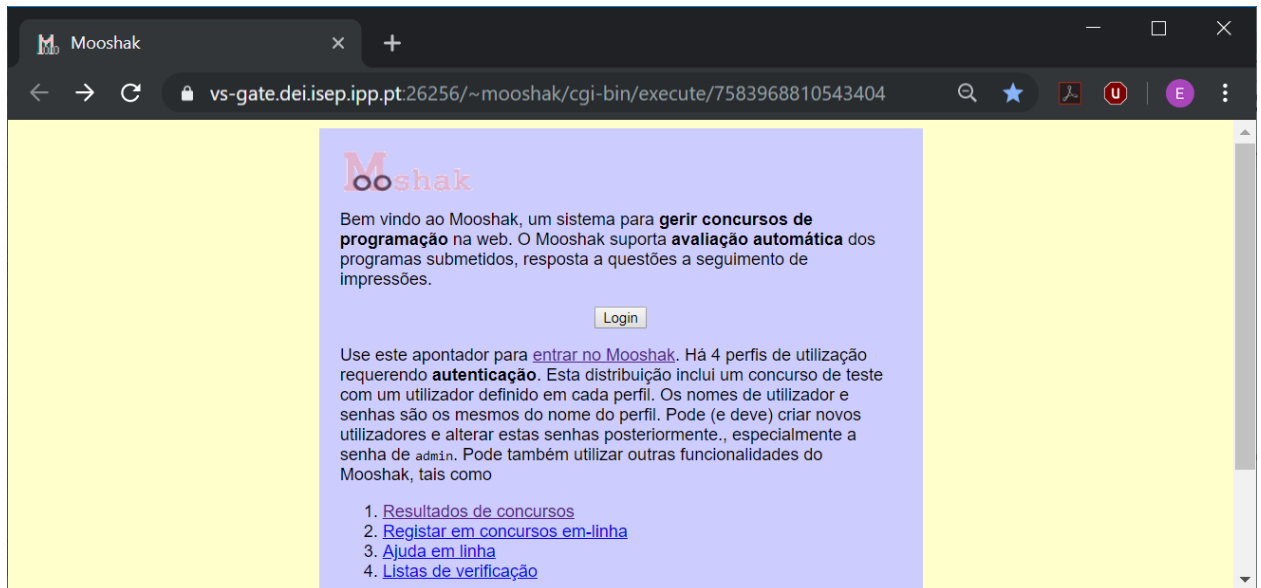
O objetivo é que os alunos resolvam os exercícios propostos e tenham um feedback instantâneo sobre as soluções submetidas. Além disso, a ferramenta disponibiliza um conjunto de estatísticas que permite ao aluno acompanhar a sua performance e evolução ao longo do tempo, bem como comparar-se com os colegas através de um ranking.

Dado um conjunto de problemas, os alunos resolvem os problemas, implementam as soluções por meio das linguagens de programação (no nosso caso será Java) e submetem o código para a ferramenta. O Mooshak recebe o código e envia-o automaticamente para compilação/execução e apresenta feedback. O código é aceite desde que satisfaça todos os casos de teste preparados previamente pelos docentes para o exercício em causa. Caso contrário, será mostrada uma mensagem de erro com base no estado de compilação/execução.

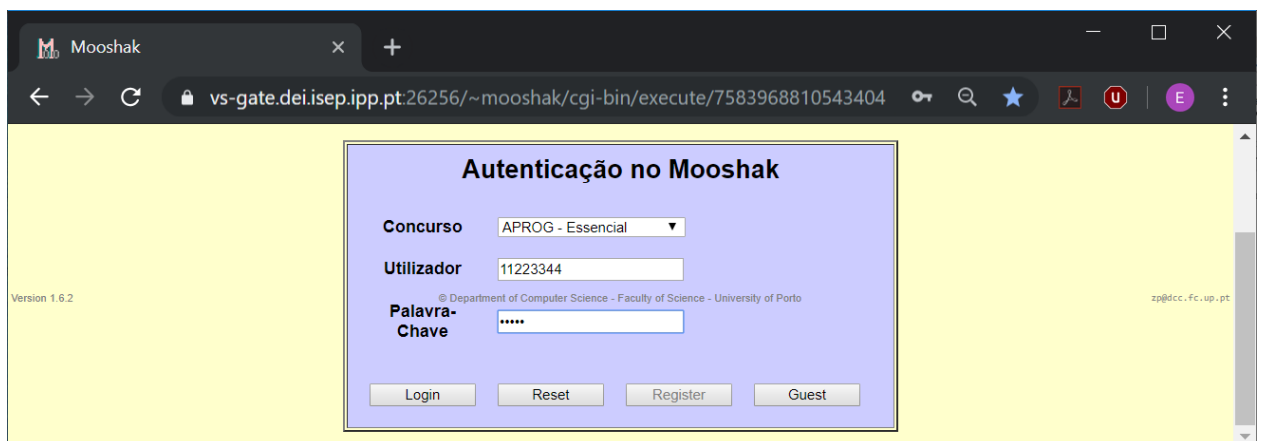
3. Funcionamento

O Mooshak a usar está disponível em:

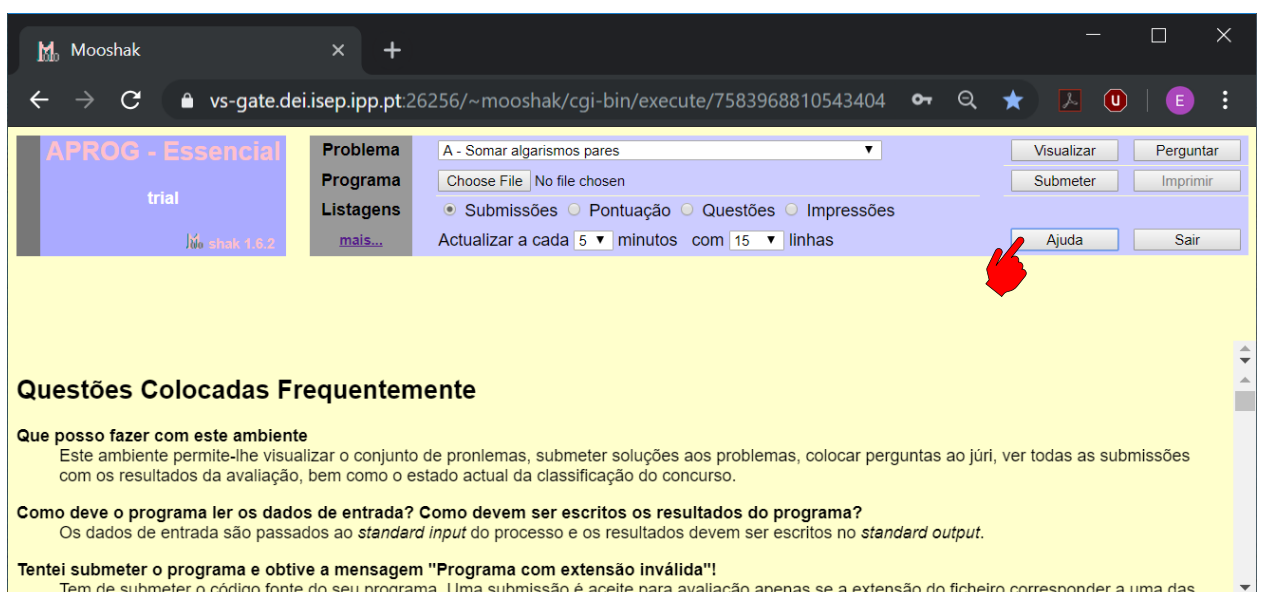
<https://vs-gate.dei.isep.ipp.pt:26256/~mooshak/cgi-bin/execute/7583968810543404>



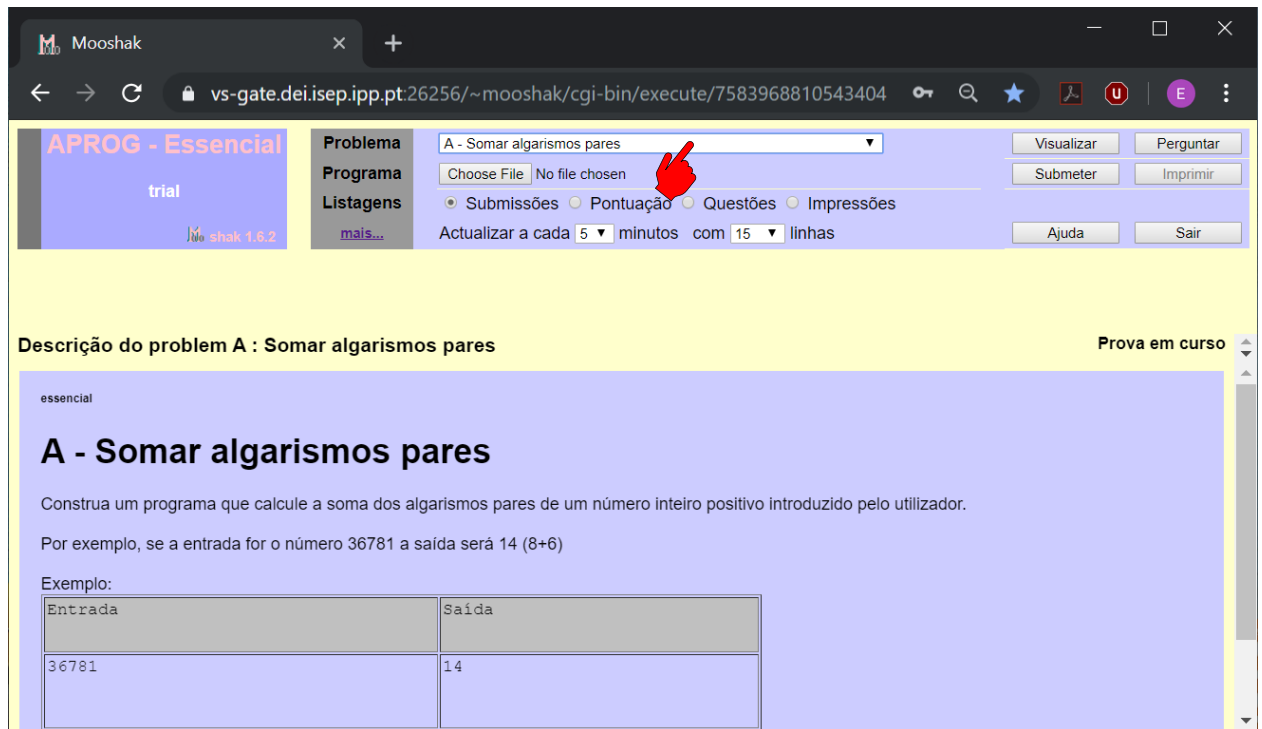
Após fazer login, será apresentado um ecrã para seleção do concurso pretendido (“Concurso”), que corresponde ao conjunto de exercícios propostos, e introdução das credenciais que lhe foram atribuídas. De seguida faz-se login.



Depois de entrar no Mooshak, é apresentada uma janela com o seguinte aspeto:



Deve começar por consultar o botão “Ajuda”. A informação disponível aborda algumas perguntas frequentes sobre este ambiente e as mensagens de feedback por ele emitidas. No topo da página, em “Problemas” está disponível a lista dos exercícios propostos para resolver. Após seleccionar um exercício e o botão “Visualizar”, é apresentada a descrição do exercício.



The screenshot shows the Mooshak web application. The top navigation bar includes 'APROG - Essencial' and 'trial'. The main content area is titled 'Descrição do problem A : Somar algarismos pares'. The problem description is as follows:

A - Somar algarismos pares

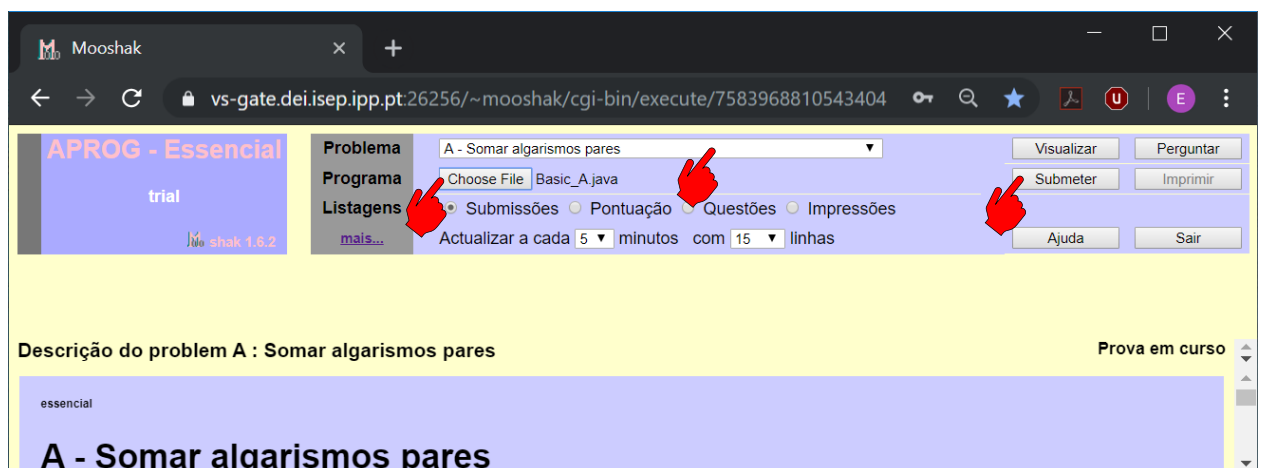
Construa um programa que calcule a soma dos algarismos pares de um número inteiro positivo introduzido pelo utilizador.

Por exemplo, se a entrada for o número 36781 a saída será 14 (8+6)

Exemplo:

Entrada	Saída
36781	14

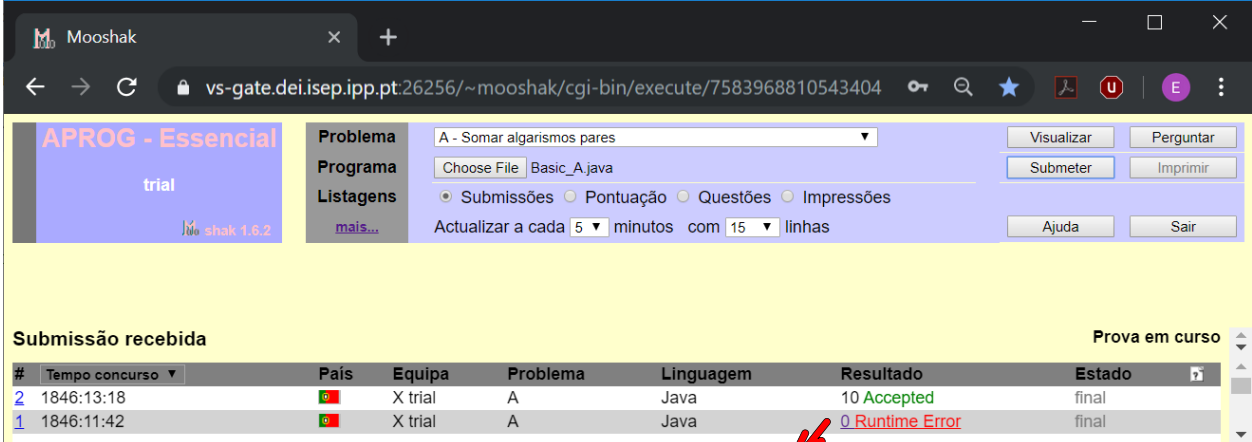
Todos os exercícios disponíveis contêm um exemplo ilustrativo do que se pretende. Para um conjunto de dados de entrada fornecido qual é a saída de dados esperada. É fundamental respeitar a formatação indicada. O exemplo fornecido ilustra o pretendido. Deve testar inicialmente o programa com o exemplo fornecido. Para submeter a resolução de um exercício, selecciona-se o problema pretendido (“Problema”), o ficheiro com o código fonte (“Programa”) e o botão “Submeter”.



The screenshot shows the Mooshak web application with the same problem description. The 'Programa' section is highlighted, showing the file 'Basic_A.java' selected. The 'Submeter' button is also visible. The problem description is repeated below the submission options.

A - Somar algarismos pares

O Mooshak recebe o programa, compila-o e executa-o com um conjunto de testes pré-definidos pelo autor do exercício. De seguida, apresenta o resultado obtido. Se o programa submetido funciona como esperado, é apresentada a mensagem “Accepted” caso contrário, é apresentada outra mensagem dependendo do tipo de erro ocorrido. Deve seleccionar o tipo de erro que aparece no “Resultado” para obter feedback adicional.

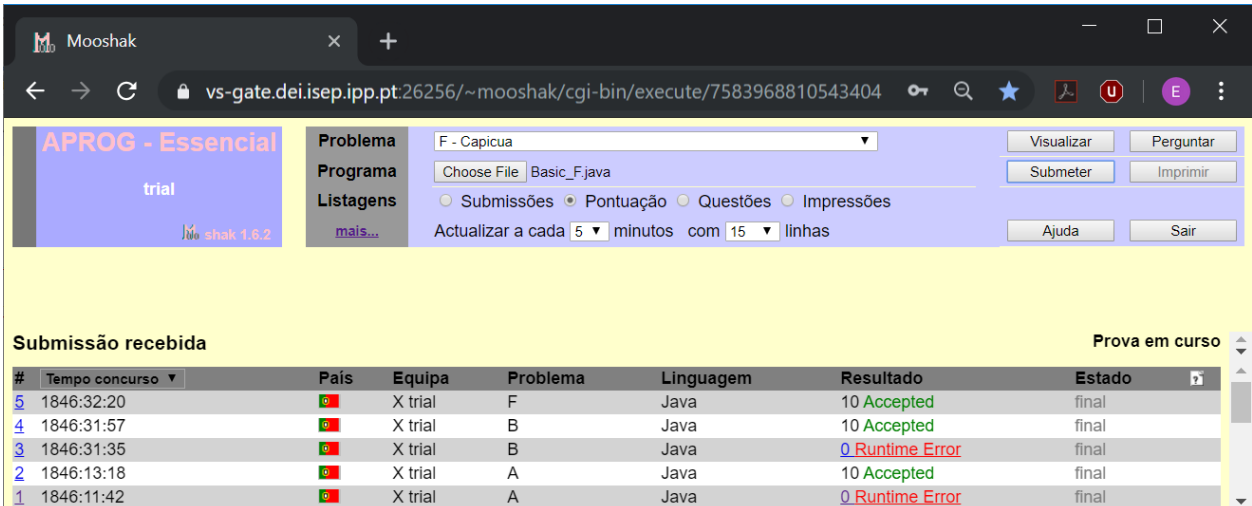


Submissão recebida

#	Tempo concurso	País	Equipa	Problema	Linguagem	Resultado	Estado
2	1846:13:18		X trial	A	Java	10 Accepted	final
1	1846:11:42		X trial	A	Java	0 Runtime Error	final

Recomenda-se, fortemente, uma leitura atenta às mensagens de erro para tentar perceber a origem do problema e encontrar uma solução adequada.

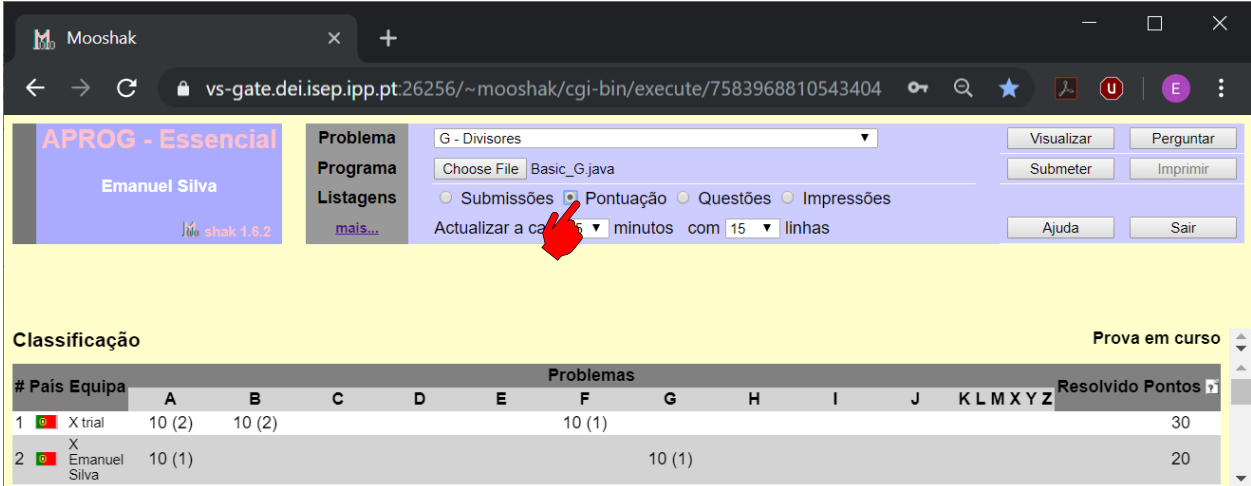
As submissões e respetivos resultados são apresentados por ordem cronológica, permitindo, desta forma, acompanhar o histórico das submissões.



Submissão recebida

#	Tempo concurso	País	Equipa	Problema	Linguagem	Resultado	Estado
5	1846:32:20		X trial	F	Java	10 Accepted	final
4	1846:31:57		X trial	B	Java	10 Accepted	final
3	1846:31:35		X trial	B	Java	0 Runtime Error	final
2	1846:13:18		X trial	A	Java	10 Accepted	final
1	1846:11:42		X trial	A	Java	0 Runtime Error	final

Ao seleccionar “Pontuação” visualiza-se uma panorâmica do desempenho global dos alunos e respetivas pontuações.



APROG - Essencial
Emanuel Silva
Mooshak 1.6.2

Problema G - Divisores
Programa Choose File Basic_G.java
Listagens Submissões **Pontuação** Questões Impressões
Actualizar a cada 5 minutos com 15 linhas

Visualizar Perguntar
Submeter Imprimir
Ajuda Sair

Classificação Prova em curso

#	País	Equipa	A	B	C	D	E	F	G	H	I	J	K	L	M	X	Y	Z	Resolvido	Pontos
1	Portugal	X trial	10 (2)	10 (2)				10 (1)											30	
2	Portugal	X Emanuel Silva	10 (1)						10 (1)										20	

Na coluna relativa a cada problema pode ver-se a pontuação obtida nesse problema e, entre parêntesis o nº de submissões realizadas. À direita é apresentada a pontuação global do aluno.

O Mooshak possui outras funcionalidades e estatísticas do desempenho pessoal e global. Assuma uma atitude ativa e curiosa e explore-as.

4. Importante

Não associe os programas a um package.

O mooshak retorna “runtime error” se isso acontecer. Neste sentido, crie, preferencialmente, um programa sem estar associado a um package. Caso contrário, e para evitar erro do mooshak, antes de submeter o exercício comente a linha referente ao package:

```
//package exercicios
public class Ex1 {
    ...
}
```

Cada programa tem de ficar integralmente incluído num único ficheiro.

No caso da linguagem Java, o ficheiro ".java" tem de ter o mesmo nome que a classe que contém o método "main". Por exemplo, o ficheiro “Ex1.java” deve ter o seguinte conteúdo:

```
public class Ex1 {
    public static void main(String[] args) {
        ...
    }
}
```

Nunca crie mais do que um Scanner para a leitura de dados.

Use sempre um único Scanner. Se precisar de ler dados em módulos distintos use um Scanner static:

```
public class Ex1 {
    static Scanner ler = new Scanner(System.in);
    public static void main(String[] args) {
        ...
    }
}
```

É necessário respeitar rigorosamente o formato exigido no enunciado.

Qualquer desacerto, mesmo ligeiro, é suficiente para que um programa seja classificado como “*Wrong Answer*” ou “*Presentation error*”. Note que não é possível detectar visualmente certas anomalias nos dados de saída. Por exemplo: um espaço em branco no fim de uma linha, uma linha em branco no fim dos dados, a omissão da mudança de linha na última linha dos dados. Todas estas situações são inaceitáveis e provocam um “*Wrong Answer*” ou “*Presentation error*”.

Existe um tempo mínimo entre submissões do mesmo exercício.

O objetivo é evitar submissões sucessivas sem sentido e sem a reflexão necessária.

Ao tentar submeter-se o mesmo problema de forma consecutiva, o Mooshak pode não permitir. Neste caso aguarde um pouco e tente novamente.