



Desafio SBPO 2025 – Regulamento

O Problema da Seleção de Pedidos Ótima

Janeiro de 2025

Conteúdo

1	Regras de participação	2
2	Organização do Desafio	3
2.1	Cronograma do Desafio	3
2.2	Conjuntos de dados	3
2.3	Avaliação	4
2.4	Função de classificação	4
2.5	Limitações de computação e de software	7
3	Formato e descrição dos arquivos	7
3.1	Entrada	7
3.2	Saída	8
4	Requisitos de código	9
4.1	Usando CPLEX	10
4.2	Usando OR-TOOLS	10
4.3	Verificador de solução	11
4.4	Programa PYTHON para a execução do desafio	11
4.5	Tempo limite de execução	11
5	Discussão e atualizações	12
6	Propriedade intelectual	12
7	Condições gerais	13

1 Regras de participação

O desafio SBPO 2025 (“Desafio”) é proposto pelo MERCADO LIVRE e tem por finalidade promover uma competição científica por meio da qual os participantes deverão desenvolver soluções algorítmicas (“Programa”) para resolver um problema de otimização que será apresentado pela empresa.

A descrição do problema-alvo do desafio (“Problema”), a disponibilização do conjunto de instâncias de *benchmark* e de um verificador de soluções, serão feitas pelo MERCADO LIVRE por meio [deste repositório](#) no GITHUB, em 15 de janeiro de 2025.

Os interessados poderão participar individualmente ou em equipe de até 3 (três) integrantes (“Equipe”), sendo que cada integrante de uma Equipe não poderá participar de outras equipes do Desafio.

A participação no Desafio é voluntária, ao fazer a inscrição todos os integrantes da Equipe declaram que leram e estão de acordo com as condições deste Regulamento, sendo certo que, caso o Participante não concorde com o quanto aqui estipulado, deverá se abster de participar do Desafio.

Todos os integrantes da Equipe devem estar, obrigatoriamente, matriculados em um programa de graduação, mestrado ou doutorado durante toda a duração do Desafio. Poderá ser exigida uma prova da vinculação dos integrantes da Equipe à instituição de ensino indicada no formulário, a qualquer momento, durante o Desafio.

Não podem participar do Desafio:

- Empregados, prepostos ou terceiros que tenham qualquer vínculo ou relação comercial com o MERCADO LIVRE; e
- Empregados e/ou terceiros que participem direta ou indiretamente da organização e condução do Desafio.

O Desafio é dividido em três etapas sucessivas: *sprint*, *qualificação* e *final*, nas quais cada Equipe poderá participar com apenas um Programa para resolver o Problema do Desafio. A participação em uma etapa não requer a participação nas etapas anteriores.

O código-fonte do Programa desenvolvido pela Equipe para solucionar o Problema do Desafio deverá estar disponibilizado, de forma pública e aberto, na plataforma GITHUB ao final de cada etapa do Desafio na qual a Equipe deseje participar. Este código-fonte será utilizado para avaliar os resultados da Equipe.

Para se inscrever no Desafio os interessados deverão preencher [este formulário](#), o qual conterá informações básicas sobre os membros da equipe e a instituição de ensino na qual estão afiliados, bem como o endereço do repositório no GITHUB onde o código-fonte com a solução proposta pela Equipe ficará localizado. O formulário deve ser preenchido *uma única vez* durante o Desafio e *antes* do término da primeira etapa na qual a Equipe deseja participar.

Participação no evento e submissão de artigo. As três equipes que obtiverem os melhores resultados no Desafio serão convidadas a apresentar seus

trabalhos durante uma sessão especial no SBPO 2025. Os membros destas Equipes estarão isentos de taxas de inscrição e são fortemente encorajados a preparar um artigo descrevendo sua proposta de solução do Problema e submetê-lo aos anais da conferência (ver a página do evento para detalhes sobre formatação e submissão).

2 Organização do Desafio

Esta seção descreve as etapas gerais do desafio.

2.1 Cronograma do Desafio

O Desafio respeitará o cronograma abaixo. Detalhes sobre os conjuntos de dados, bem como sobre a disponibilização de códigos-fonte que estão mencionados no cronograma são fornecidos nas subseções seguintes.

- a) 17/01/2025: Lançamento oficial do Desafio e início da *etapa sprint*. Neste momento o Problema será apresentado para os participantes. O conjunto de dados A estará disponível.
- b) 30/03/2025: Término da etapa *sprint* e data limite para inclusão dos códigos-fonte correspondentes a esta etapa no GITHUB;
- c) 15/04/2025: Início da *etapa de qualificação* e divulgação dos resultados apurados na etapa impressão. O conjunto de dados B estará disponível;
- d) 16/06/2025: Término da etapa de qualificação e data limite para inclusão dos códigos fontes correspondentes a esta etapa no GITHUB;
- e) 30/06/2025: Início da etapa final. Publicação da classificação das Equipes na etapa de qualificação;
- f) 01/09/2025: Término da etapa final e data limite para inclusão dos códigos fontes correspondentes a esta etapa no GITHUB;
- g) 15/09/2025: Anúncio parcial das classificações finais, excluindo as três primeiras posições. O conjunto de dados X estará disponível; e
- h) Entre 6 a 09 de outubro de 2025, de acordo com a programação da SBPO 2025: Anúncio da equipe vencedora.

2.2 Conjuntos de dados

Serão fornecidos às Equipes 3 (três) conjuntos de dados baseados em instâncias reais do problema durante o Desafio:

- Conjunto de Dados A – 20 instâncias disponíveis no lançamento do Desafio;

- Conjunto de Dados B – 15 instâncias disponíveis ao término da etapa de qualificação; e
- Conjunto de Dados X – 15 instâncias ocultas que serão usadas para classificar as equipes na etapa final e que estará disponível após o término do Desafio.

O conjunto de dados A contém instâncias de tamanho pequeno/médio, enquanto os conjuntos B e X contém instâncias de tamanho regular.

2.3 Avaliação

No prazo final de cada etapa, as equipes devem disponibilizar seus programas no GITHUB (consulte a Seção 2.5 para as limitações de software).

Para fins de comparação entre as Equipes:

1. O conjunto de dados A será usado na etapa de *sprint* e ao final da etapa de qualificação;
2. O conjunto de dados B será usado nas etapas de qualificação e final; e
3. O conjunto de dados X será usado exclusivamente na etapa final.

O programa de cada Equipe será executado com um limite de tempo de 10 minutos. Seja Z uma instância do problema em um conjunto de dados e $V(Z)$ o custo da solução encontrada para a instância Z dentro do limite de tempo. O valor de $V(Z)$ será usado para determinar a pontuação da Equipe, conforme descrito a seguir.

2.4 Função de classificação

No final de cada etapa as Equipes serão classificadas seguindo a função de classificação descrita abaixo.

Cada Equipe recebe pontos para cada instância, dependendo de seu resultado. Dada uma instância Z ,

- O termo *pontuação* refere-se ao número de pontos recebidos por uma Equipe em relação a Z .
- O termo *resultado* refere-se ao valor objetivo obtido pela Equipe, ou seja, $V(Z)$.

A Equipe vencedora será aquela que maximizar a soma de suas pontuações em todas as instâncias.

Suponha, então, que m seja o número total de instâncias em um conjunto de dados e considere que o problema tratado esteja na forma de maximização. Seja y_{ij} o resultado da Equipe i para a instância j quando a solução é viável, e $y_{ij} = -\infty$ caso contrário ou se o programa da Equipe travar para essa instância

j . Além disso, seja $n_{\text{better}}(ij)$ o número de equipes com resultado estritamente melhor que o da Equipe i na instância j , isto é,

$$n_{\text{better}}(ij) = \left| \{k \neq i \text{ tal que } y_{kj} > y_{ij}\} \right|.$$

Seja R uma constante que representa a pontuação máxima que uma equipe pode obter em uma instância específica de uma etapa da competição (o valor de R para o desafio atual será fornecido posteriormente neste documento). O número de pontos obtidos pela Equipe i (ou seja, a sua pontuação da Equipe i) na instância j nesta etapa é definida como:

$$p_{ij} = \begin{cases} \max\{0, R - n_{\text{better}}(ij)\}, & \text{se a solução for viável,} \\ 0, & \text{se nenhuma solução for encontrada} \\ -1, & \text{ou se a solução for inviável,} \\ & \text{se o programa travar.} \end{cases} \quad (1)$$

A pontuação global de uma Equipe, denotada por $\text{score}(i)$, é definida por:

$$\text{score}(i) = \sum_{j \in [1, m]} p_{ij}.$$

A quantidade de *melhores soluções* obtidas pelas Equipes será usada na tentativa de quebrar eventuais empates nas pontuações. Neste contexto, dada uma instância Z , a solução de uma Equipe i será dita a *melhor* para Z se nenhuma outra equipe tiver alcançado um valor objetivo estritamente maior do que aquele obtido pela Equipe i para esta instância. Portanto, em caso de empate, quanto maior o número de melhores soluções encontradas por uma Equipe, melhor será a sua classificação em relação a outras Equipes com a mesma pontuação.

A seguir ilustra-se como a Equipe vencedora é identificada. Considere o exemplo abaixo com $m = 6$ instâncias.

Instância	1	2	3	4	5	6
Equipe 1	66	65	58	68	90	88
Equipe 2	68	76	56	67	87	85
Equipe 3	67	64	70	45	90	83
Equipe 4	64	68	54	68	88	87
Equipe 5	63	70	57	71	91	96
Equipe 6	inviável	71	59	88	90	95
Equipe 7	61	70	57	travou	90	96

Tabela 1: Pontuações das Equipes para diferentes instâncias (valores de y_{ij}).

Os valores de $n_{\text{better}}(ij)$ são os seguintes:

Instância	1	2	3	4	5	6
Equipe 1	2	5	2	2	1	3
Equipe 2	0	0	5	4	6	5
Equipe 3	1	6	0	5	1	6
Equipe 4	3	4	6	2	5	4
Equipe 5	4	2	3	1	0	0
Equipe 6	6	1	1	0	1	2
Equipe 7	5	2	3	6	1	0

Tabela 2: Valores de $n_{\text{better}}(ij)$.

Supondo $R = 5$, as pontuações p_{ij} são as seguintes:

Instância	1	2	3	4	5	6
Equipe 1	3	0	3	3	4	2
Equipe 2	5	5	0	1	0	0
Equipe 3	4	0	5	0	4	0
Equipe 4	2	1	0	3	0	1
Equipe 5	1	3	2	4	5	5
Equipe 6	0	4	4	5	4	3
Equipe 7	0	3	2	-1	4	5

Tabela 3: Valores de p_{ij} para $R = 5$.

Por fim, as pontuações globais são:

Equipe	Pontuação Final
Equipe 1	15
Equipe 2	11
Equipe 3	13
Equipe 4	7
Equipe 5	20
Equipe 6	20
Equipe 7	13

Tabela 4: Valores de $\text{score}(i)$ para $R = 5$.

Neste exemplo, a Equipe 5 seria a vencedora. Ela possui a mesma pontuação que a Equipe 6 (20), mas um número maior de melhores soluções conhecidas (2 contra 1).

Em qualquer etapa do Desafio SBPO 2025, o valor de R será definido como o número de Equipes participantes na respectiva etapa.

2.5 Limitações de computação e de software

Para garantir uma comparação justa dos métodos desenvolvidos pelas diferentes Equipes participantes, cada Equipe deve submeter seu código fonte em todas as etapas. Este programa deve aceitar uma instância de problema de um dos conjuntos de dados, seguindo o formato especificado na Seção 3.1. Ele deve produzir um arquivo de solução de saída de acordo com os padrões definidos na Seção 3.2. Para cada instância, os organizadores farão *uma única* execução usando uma semente fixa escolhida aleatoriamente.

Para promover transparência e justiça, a semente selecionada pelos organizadores para avaliações e classificações das equipes será divulgada após a conclusão de cada etapa. Essa semente será aplicada consistentemente para cada equipe e cada instância. É responsabilidade dos participantes garantir que o valor da semente seja passado como um parâmetro para o programa executável, possibilitando assim a repetibilidade dos experimentos e avaliações, especialmente se seus métodos de solução envolverem estruturas ou componentes probabilísticos. Os programas serão avaliados apenas *uma única vez* por instância; *nenhuma avaliação adicional será conduzida*. Caso ocorra variabilidade, os organizadores não podem ser responsabilizados por quaisquer efeitos adversos resultantes dessa única execução.

O computador que será usado para avaliar os programas dos candidatos é uma máquina com sistema operacional Linux, que permite multi-threading e até 64GB de RAM. *O uso de GPUs não estará disponível.*

As equipes que desejarem usar um resolvidor de Programação Inteira (ILP) devem utilizar o CPLEX 22.11 ou o OR-TOOLS 4.11. Os códigos devem ser escritos em JAVA. Nenhuma outra linguagem será aceita.

3 Formato e descrição dos arquivos

Esta seção descreve o formato dos arquivos usados no desafio. Para uma melhor compreensão do material apresentado aqui, recomenda-se ler primeiro o *arquivo de descrição do problema*, que está disponível [neste repositório](#).

3.1 Entrada

A entrada será fornecida em um arquivo de texto com o seguinte formato:

- Primeira linha: contém três inteiros o , i e a , que representam, respectivamente, o número de pedidos, itens e corredores.
- Próximas o linhas: cada linha começa com um inteiro k , seguido por k pares de inteiros, representando o número do item e o número de unidades solicitadas pelo pedido. Dentro deste grupo de linhas e , para j em $\{1, \dots, o\}$, a j -ésima linha contém dados relacionados ao pedido indexado por $j - 1$, significando que os pedidos são numerados de 0 a $o - 1$. Além disso, assume-se que os itens estão indexados de 0 a $i - 1$.

- Próximas a linhas: cada linha começa com um inteiro l , seguido por l pares de inteiros, representando o número do item e o número de unidades disponíveis no corredor. Similar ao caso dos pedidos, dentro deste grupo de linhas e, para j em $\{1, \dots, a\}$, a j -ésima linha contém dados relacionados ao corredor indexado por $j - 1$, significando que os corredores são numerados de 0 a $a - 1$.
- Última linha: contém dois inteiros, LB e UB, que são os limites inferior e superior do tamanho da *wave* (um subconjunto de pedidos), respectivamente. Esses limites são expressos em quantidade de unidades de itens.

Assim, a entrada para a instância descrita na *Seção de Exemplos* do [arquivo de descrição do problema](#) seria:

```

5 5 5
2 0 3 2 1
2 1 1 3 1
2 2 1 4 2
4 0 1 2 2 3 1 4 1
1 1 1
4 0 2 1 1 2 1 4 1
4 0 2 1 1 2 2 4 1
3 1 2 3 1 4 2
4 0 2 1 1 3 1 4 1
4 1 1 2 2 3 1 4 2
5 12

```

3.2 Saída

A saída deve ser um arquivo de texto com o seguinte formato:

- Primeira linha: contém um inteiro n , o número de pedidos na *wave* (subconjunto de pedidos) retornada pelo algoritmo.
- Próximas n linhas: cada linha contém um inteiro representando o índice de um pedido da *wave* computada.
- Linha seguinte: contém um inteiro m que representa o número de corredores visitados na *wave* computada.
- Próximas m linhas: cada linha contém um inteiro representando o índice de um corredor visitado pela *wave* computada.

Por exemplo, a saída para a última solução do exemplo descrito na Seção 3 do [arquivo de descrição do problema](#) seria:

```

4
0

```


1
2
4
2
1
3

4 Requisitos de código

Como mencionado na Seção 2.5, o código deve ser escrito em JAVA, especificamente, na versão 17. Para facilitar o processo de desenvolvimento e avaliação, fornecemos um *template* para o projeto a ser desenvolvido no GITHUB que pode ser clonado [este repositório](#).

Uma vez que o projeto tenha sido clonado para seu repositório local, o diretório raiz, chamado **ChallengeSBPO2025** (**não o altere!**), conterá os seguintes elementos:

`checker.py run_challenge.py pom.xml src`

O arquivo `checker.py` é um script PYTHON que verifica se uma solução é viável (veja a Seção 4.3), enquanto o script `run_challenge.py` pode facilmente compilar e executar um código JAVA, em conformidade com as especificações deste documento, sobre um conjunto de instâncias. Como **MAVEN** será usado para construir o projeto, o `pom.xml` (Project Object Model) também é fornecido. Ele contém informações sobre o projeto e detalhes de configuração usados pelo MAVEN. **Não altere este arquivo, exceto se o CPLEX ou o OR-TOOLS forem requeridos pelo código desenvolvido para o desafio** (veja as Seções 4.1 e 4.2 abaixo). O diretório `src` é onde os códigos JAVA devem residir.

Após clonar o projeto disponibilizado no GITHUB, seguindo o caminho `/src/main/java/org/sbpo2025/challenge/` três arquivos serão encontrados:

1. `Challenge.java`. Este é o programa *main* do projeto e define a classe `Challenge` que contém métodos para ler o arquivo de entrada e escrever o arquivo de saída.
2. `ChallengeSolver.java`. Este arquivo contém um código incompleto **que deve ser preenchido pelas Equipes para implementar os algoritmos propostos**. Ele define a classe `ChallengeSolver` que é chamada por `Challenge.java` e inclui algumas funções auxiliares para verificar a viabilidade e calcular o valor objetivo de uma solução candidata.
3. `ChallengeSolution.java`. Este arquivo define uma estrutura básica que representa uma solução para o problema.

Embora seja altamente recomendado, as equipes não são obrigadas a manter o código em `Challenge.java` inalterado. No entanto, independentemente de

quaisquer modificações feitas, **o arquivo executável resultante deve aceitar dois argumentos: os caminhos dos arquivos de entrada e saída, nessa ordem.**

O projeto pode ser compilado usando o comando MAVEN emitido a partir do diretório raiz:

```
mvn clean package
```

Isso gerará um arquivo JAR, chamado `ChallengeSBPO2025-1.0.jar`, no diretório `target`.

Para executar o arquivo JAR, o seguinte comando pode ser emitido a partir do diretório raiz do projeto:

```
java -jar target/ChallengeSBPO2025-1.0.jar  
<input-file> <output-file>
```

onde `<input-file>` é o caminho para o arquivo de entrada a ser lido, e `<output-file>` é o caminho para o arquivo de saída a ser escrito.

4.1 Usando CPLEX

Se o CPLEX for utilizado pelo algoritmo proposto, todas as linhas do *bloco de dependências* do arquivo `pom.xml` devem ser descomentadas. Além disso, antes de compilar o código, as dependências do CPLEX devem ser adicionadas ao repositório MAVEN usando o seguinte comando:

```
mvn install:install-file -Dfile=<cplex-jar-file>  
-DgroupId=cplex -DartifactId=cplex  
-Dversion=22.11 -Dpackaging=jar
```

onde `<cplex-jar-file>` é o caminho para o arquivo CPLEX JAR (por exemplo, `/Applications/CPLEX_Studio2211/cplex/lib/cplex.jar`). Observe que **a versão 22.11** do resolvidor será utilizada para avaliação.

Depois disso, para executar o arquivo de aplicação JAR com o CPLEX, você deve fornecer o caminho para a biblioteca do CPLEX, usando o seguinte comando:

```
java -Djava.library.path=<cplex-library-path>  
-jar target/ChallengeSBPO2025-1.0.jar  
<input-file> <output-file>
```

onde `<cplex-library-path>` é o caminho para a biblioteca do CPLEX na sua instalação local (por exemplo, `/Applications/CPLEX_Studio2211/opl/bin/arm64_osx/`).

4.2 Usando OR-TOOLS

Se você decidir usar OR-TOOLS, deve primeiro instalar OR-TOOLS para Java, seguindo as instruções [nesta página](#). Observe que **a versão 4.11** da biblioteca

será utilizada para avaliação. Em seguida, você deve descomentar a respectiva dependência no arquivo `pom.xml`.

Finalmente, você pode executar o arquivo JAR com OR-TOOLS usando o seguinte comando:

```
java -Djava.library.path=<ortools-library-path>
-jar target/ChallengeSBPO2025-1.0.jar
<input-file> <output-file>
```

onde `<ortools-library-path>` é o caminho para a biblioteca OR-TOOLS (por exemplo, `$HOME/Documents/or-tools/build/lib/`).

4.3 Verificador de solução

Um script PYTHON, chamado `checker.py`, é fornecido para verificar a viabilidade de uma solução com base nos arquivos de entrada e saída passados como parâmetros, além de calcular o valor objetivo. O script pode ser executado usando o seguinte comando:

```
python3 checker.py <input-file> <output-file>
```

4.4 Programa PYTHON para a execução do desafio

O script `run_challenge.py` pode ser usado para compilar e executar o código JAVA para um conjunto de instâncias. Ele requer o caminho do código fonte (que contém o arquivo `pom.xml`), o diretório de entrada (instâncias) e o diretório de saída (para armazenar as soluções) como parâmetros, conforme mostrado abaixo:

```
python3 run_challenge.py <source-code-path>
                        <input-directory>
                        <output-directory>
```

Para executar este script, você precisará do comando `timeout` (ou `gtimeout` no macOS) instalado. Você pode instalá-lo usando o comando `apt-get install coreutils` (ou equivalente) no Linux ou `brew install coreutils` no macOS. O script permite que o usuário configure os caminhos para as bibliotecas do CPLEX e do OR-TOOLS, além de interromper automaticamente a execução do programa caso o limite de tempo de 10 minutos seja excedido.

4.5 Tempo limite de execução

Lembre-se de que um limite de tempo de 10 minutos será alocado para computar uma solução para cada instância. Portanto, recomenda-se usar algum pacote JAVA, como [StopWatch](#), para monitorar o tempo de execução e garantir que um arquivo de saída seja escrito *dentro deste limite de tempo*.

5 Discussão e atualizações

Para ajudar as Equipes a gerenciar seus projetos, os organizadores estão se comprometendo a comunicar quaisquer alterações que possam ocorrer durante o desafio com a máxima brevidade. Isso inclui esclarecimentos sobre a descrição do problema ou alterações nas instâncias do problema, por exemplo. Essas mudanças serão feitas apenas se forem estritamente necessárias e serão comunicadas através do [GITHUB do repositório](#) do desafio. Os participantes podem entrar em contato com os organizadores através do endereço de e-mail sor-challenges@mercadolibre.com, mas apenas em caso de dúvidas ou problemas significativos.

6 Propriedade intelectual

As equipes participantes do desafio manterão os direitos autorais sobre os programas de computador desenvolvidos durante o desafio. O MERCADO LIVRE e quaisquer terceiros podem usar informações fornecidas pelos participantes por meio de relatórios técnicos, artigos científicos e apresentações orais, mas não poderão usar um programa de computador sem o acordo da equipe que desenvolveu esse programa.

Os participantes do desafio não podem afirmar que têm uma parceria com o MERCADO LIVRE ou qualquer contrato com o MERCADO LIVRE ou com entidades do grupo MERCADO LIVRE, mesmo que vençam o desafio. Eles podem apenas alegar que participaram (respectivamente, foram qualificados ou vencedores) do desafio, caso seja o caso.

7 Condições gerais

Todas as decisões sobre o Desafio, inclusive, esclarecimento de eventuais dúvidas sobre a mecânica deste projeto ficam sob responsabilidade do MERCADO LIVRE.

Os Participantes reconhecem e concordam expressamente que o Desafio de excelência **não** estabelece qualquer vínculo empregatício ou remuneração salarial.

O MERCADO LIVRE reserva-se ao direito de alterar o presente Regulamento a qualquer momento, no todo ou em parte, mediante prévio aviso aos Participantes, através de comunicado por escrito, sendo certo que a continuidade do Participante no Desafio, após tais modificações, será presumida como aceitação tácita das novas regras.

A participação no Desafio não gerará ao Participante nenhum outro direito ou vantagem que não esteja expressamente previsto neste Regulamento.

Os Participantes, no ato de seu cadastro, autorizam o MERCADO LIVRE a utilizar, para fins administrativos e de marketing, todos os seus dados cadastrais e as informações de consumo geradas, desde que respeitadas as disposições legais vigentes.

As dúvidas eventualmente existentes neste Regulamento serão tratadas diretamente entre o participante e o MERCADO LIVRE e poderão ser encaminhadas pelo e-mail de contato: sor-challenges@mercadolibre.com.

O Desafio será regulado exclusivamente por este Regulamento, que pode ser substituído ou alterado a critério único e exclusivo do MERCADO LIVRE. No caso de descredenciamento de um integrante da Equipe da instituição de estudo indicada no cadastro, a Equipe será desclassificada.

Na hipótese de verificação de fraude, tentativa ou suspeita de fraude de uma Equipe, a pontuação atribuída à Equipe será zerada. Uma vez confirmada a fraude, a Equipe será automaticamente excluída do Desafio, sem prejuízo das medidas legais cabíveis.

O MERCADO LIVRE reserva-se o direito de decidir sobre qualquer ponto não especificado neste Regulamento, sendo sua decisão soberana e incontestável.

Casos omissos e/ou não previstos neste Regulamento serão de boa-fé, resolvidos pela comissão do MERCADO LIVRE responsável pelo desafio.

Os dados coletados no Desafio são utilizados para regular o gerenciamento do Desafio e ações nele realizadas, identificação dos Usuários, comunicação, acompanhamento de desempenho, disponibilização de extratos, elaboração de relatórios e demais análises aplicáveis, inclusive para fins estatísticos.

O MERCADO LIVRE compromete-se a respeitar a confidencialidade de todos e quaisquer dados e informações relativas a uma pessoa física por meio da qual a mesma possa ser identificada ou identificável (“Dados Pessoais”) ou comerciais eventualmente coletados através da internet, independentemente do sistema de coleta utilizado, sejam dos Participantes ou de quaisquer terceiros (os “Dados Eletrônicos”), observando, para tanto, a Lei nº 13.709/2018 (Lei Geral de Proteção de Dados Pessoais), a Lei nº 12.965/2014 (Marco Civil da Internet) e Decreto nº 8.771/2016 (Regulamento do Marco Civil da Internet), bem como quaisquer outras leis relativas à proteção de dados pessoais obtidos pela internet

que vierem a ser promulgadas na República Federativa do Brasil ou entrarem em vigor no curso da vigência deste Contrato.

O uso das informações pessoais dos participantes do Desafio como nome, CPF, e-mail e whatsapp, informações estas que servem para tornar identificáveis todos que participam do Desafio durante a sua vigência, se darão de acordo com a legislação, especialmente a Lei nº 13.709/18 sendo que os processos de tratamento somente poderão ser realizados para propósitos legítimos, específicos, explícitos e previamente informados aos titulares dos Dados Pessoais.

O participante, ao aceitar esse regulamento, aceita o tratamento dos seus dados pessoais pelo MERCADO LIVRE.

O MERCADO LIVRE garante que adotará medidas de prevenção de todos os dados pessoais dos participantes cadastrados na campanha.

Todos os dados pessoais dos participantes serão usados única e exclusivamente em decorrência do Desafio, estando, então, vetado o uso e/ou tratamento desses Dados Pessoais sem o prévio consentimento de seus titulares, para finalidades diferentes das expressamente determinadas neste regulamento.

O MERCADO LIVRE e o participante acordam juntos que, após o término da vigência da campanha, todos os Dados Pessoais cadastrados serão eliminados.

Fica eleito o Foro da Comarca de São Paulo/SP para quaisquer disputas judiciais que, eventualmente, venham a existir em virtude das disposições previstas no presente Regulamento.