Desafío SBPO 2025 – Reglamento

El Problema de la Selección de Órdenes Óptima

Enero de 2025

Conteúdo

1 Reglas de participación

El desafío SBPO 2025 ("Desafío") es propuesto por MERCADO LIVRE y tiene como finalidad promover una competencia científica a través de la cual los participantes deberán desarrollar soluciones algorítmicas ("Programa") para resolver un problema de optimización que será presentado por la empresa.

La descripción del problema objetivo del desafío ("Problema"), la disponibilización del conjunto de instancias de *benchmark* y de un verificador de soluciones, serán realizadas por MERCADO LIVRE a través de este repositorio en GITHUB, el 15 de enero de 2025.

Los interesados podrán participar individualmente o en equipo de hasta 3 (tres) integrantes ("Equipo"), siendo que cada integrante de un Equipo no podrá participar en otros equipos del Desafío.

La participación en el Desafío es voluntaria, al momento de la inscripción todos los integrantes del Equipo declaran que han leído y están de acuerdo con las condiciones de este Reglamento, siendo cierto que, en caso de que el Participante no concorde con lo estipulado aquí, deberá abstenerse de participar en el Desafío.

Todos los integrantes del Equipo deben estar, obligatoriamente, matriculados en un programa de grado, maestría o doctorado en una institución educativa de un país donde MERCADO LIVRE tenga operaciones, y esto durante toda la duración del Desafío. Se podrá exigir una prueba de vinculación de los integrantes del Equipo a la institución de educación indicada en el formulario, en cualquier momento, durante el Desafío.

No pueden participar en el Desafío:

- Empleados, representantes o terceros que tengan cualquier vínculo o relación comercial con MERCADO LIVRE; y
- Empleados y/o terceros que participen directa o indirectamente de la organización y conducción del Desafío.

El Desafío está dividido en tres etapas sucesivas: *sprint*, *calificación* y *final*, en las cuales cada Equipo podrá participar con *solo un* Programa para resolver el Problema del Desafío. La participación en una etapa no requiere la participación en las etapas anteriores.

El código fuente del Programa desarrollado por el Equipo para solucionar el Problema del Desafío deberá estar disponible, de forma pública y abierta, en la plataforma GITHUB al final de cada etapa del Desafío en la que el Equipo desee participar. Este código fuente será utilizado para evaluar los resultados del Equipo.

Para inscribirse en el Desafío, los interesados deberán llenar este formulario, el cual contendrá información básica sobre los miembros del equipo y la institución de educación a la que están afiliados, así como la dirección del repositorio en GITHUB donde el código fuente con la solución propuesta por el Equipo quedará ubicado. El formulario debe ser llenado *una única vez* durante el Desafío y *antes* del término de la primera etapa en la que el Equipo desea participar.

Participación en el evento y presentación de artículo. Los tres equipos que obtengan los mejores resultados en el Desafío serán invitados a presentar sus trabajos durante una sesión especial en SBPO 2025. Los miembros de estos Equipos estarán exentos de tasas de inscripción y son fuertemente alentados a preparar un artículo describiendo su propuesta de solución del Problema y presentarlo a los anales de la conferencia (ver la página del evento para detalles sobre formateo y presentación).

2 Organización del Desafío

Esta sección describe las etapas generales del desafío.

2.1 Cronograma del Desafío

El Desafío respetará el cronograma a continuación. Detalles sobre los conjuntos de datos, así como sobre la disponibilización de códigos fuente que están mencionados en el cronograma, se proporcionan en las subsecciones siguientes.

a) 17/01/2025: Lanzamiento oficial del Desafío y comienzo de la *etapa sprint*. En este momento, el Problema será presentado a los participantes. El conjunto de datos A estará disponible.

b) 30/03/2025: Término de la etapa *sprint* y fecha límite para inclusión de los códigos fuente correspondientes a esta etapa en GITHUB;

c) 15/04/2025: Inicio de la *etapa de calificación* y divulgación de los resultados obtenidos en la etapa anterior. El conjunto de datos B estará disponible;

d) 16/06/2025: Término de la etapa de calificación y fecha límite para inclusión de los códigos fuentes correspondientes a esta etapa en GITHUB;

e) 30/06/2025: Inicio de la etapa final. Publicación de la clasificación de los Equipos en la etapa de calificación;

f) 01/09/2025: Término de la etapa final y fecha límite para inclusión de los códigos fuentes correspondientes a esta etapa en GITHUB;

g) 15/09/2025: Anuncio parcial de las clasificaciones finales, excluyendo las tres primeras posiciones. El conjunto de datos X estará disponible; y

h) Entre el 6 y el 09 de octubre de 2025, de acuerdo con la programación de SBPO 2025: Anuncio del equipo vencedor.

2.2 Conjuntos de datos

Se proporcionarán a los Equipos 3 (tres) conjuntos de datos basados en instancias reales del problema durante el Desafío:

- Conjunto de Datos A – 20 instancias disponibles en el lanzamiento del Desafío;
- Conjunto de Datos B – 15 instancias disponibles al término de la etapa de calificación; y
- Conjunto de Datos X – 15 instancias ocultas que se utilizarán para clasificar a los equipos en la etapa final y que estarán disponibles tras el término del Desafío.

El conjunto de datos A contiene instancias de tamaño pequeño/medio, mientras que los conjuntos B y X contienen instancias de tamaño regular.

2.3 Evaluación

Al final de cada etapa, los equipos deben poner a disposición sus programas en GITHUB (consulte la Sección 2.5 para las limitaciones de software).

Para fines de comparación entre los Equipos:

1. El conjunto de datos A se utilizará en la etapa de *sprint* y al final de la etapa de calificación;
2. El conjunto de datos B se utilizará en las etapas de calificación y final; y
3. El conjunto de datos X se usará exclusivamente en la etapa final.

El programa de cada Equipo se ejecutará con un límite de tiempo de 10 minutos. Sea Z una instancia del problema en un conjunto de datos y $V(Z)$ el costo de la solución encontrada para la instancia Z dentro del límite de tiempo. El valor de $V(Z)$ se utilizará para determinar la puntuación del Equipo, conforme a lo descrito a continuación.

2.4 Función de clasificación

Al final de cada etapa, los Equipos serán clasificados siguiendo la función de clasificación descrita a continuación.

Cada Equipo recibe puntos para cada instancia, dependiendo de su resultado. Dada una instancia Z,

- El término *puntuación* se refiere al número de puntos recibidos por un Equipo en relación a Z.
- El término *resultado* se refiere al valor objetivo obtenido por la Equipo, es decir, $V(Z)$.

El Equipo vencedor será aquel que maximice la suma de sus puntuaciones en todas las instancias.

Supongamos, entonces, que m sea el número total de instancias en un conjunto de datos y consideremos que el problema tratado esté en forma de maximización. Sea y_{ij} el resultado del Equipo i para la instancia j cuando la solución es factible, y $y_{ij} = -\infty$ en caso contrario o si el programa del Equipo falla para esa instancia j. Además, sea $n_{\text{better}}(ij)$ el número de equipos con resultado estrictamente mejor que el de Equipo i en la instancia j, es decir,

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

Sea R una constante que representa la puntuación máxima que un equipo puede obtener en una instancia específica de una etapa de la competencia (el valor de R para el desafío actual será proporcionado posteriormente en este documento). El número de puntos obtenidos por el Equipo i (es decir, su puntuación del Equipo i) en la instancia j en esta etapa se define como:

$$p_{ij} = \begin{cases} \max\{0, R - n_{\text{better}}(ij)\}, & \text{si la solución es factible,} \\ 0, & \text{si no se encuentra ninguna solución} \\ & \text{o si la solución es infactible,} \\ -1, & \text{si el programa falla.} \end{cases} \tag{1}$$

La puntuación global de un Equipo, denotada por score(i), se define por:

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

La cantidad de *mejores soluciones* obtenidas por los Equipos se utilizará en un intento de romper eventuales empates en las puntuaciones. En este contexto, dada una instancia Z, la solución de un Equipo i se considerará la *mejor* para Z si ningún otro equipo ha alcanzado un valor objetivo estrictamente mayor que el obtenido por el Equipo i para esta instancia. Por lo tanto, en caso de empate, cuanto mayor sea el número de mejores soluciones encontradas por un Equipo, mejor será su clasificación en relación a otros Equipos con la misma puntuación.

A continuación se ilustra cómo se identifica al Equipo vencedor. Considere el siguiente ejemplo con $m = 6$ instancias.

Los valores de $n_{\text{better}}(ij)$ son los siguientes:

Instancia	1	2	3	4	5	6
Equipo 1	66	65	58	68	90	88
Equipo 2	68	76	56	67	87	85
Equipo 3	67	64	70	45	90	83
Equipo 4	64	68	54	68	88	87
Equipo 5	63	70	57	71	91	96
Equipo 6	infactible	71	59	88	90	95
Equipo 7	61	70	57	falló	90	96

Tabela 1: Puntuaciones de los Equipos para diferentes instancias (valores de y_{ij}).

Instancia	1	2	3	4	5	6
Equipo 1	2	5	2	2	1	3
Equipo 2	0	0	5	4	6	5
Equipo 3	1	6	0	5	1	6
Equipo 4	3	4	6	2	5	4
Equipo 5	4	2	3	1	0	0
Equipo 6	6	1	1	0	1	2
Equipo 7	5	2	3	6	1	0

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

Suponiendo $R = 5$, las puntuaciones p_{ij} son las siguientes:

Instancia	1	2	3	4	5	6
Equipo 1	3	0	3	3	4	2
Equipo 2	5	5	0	1	0	0
Equipo 3	4	0	5	0	4	0
Equipo 4	2	1	0	3	0	1
Equipo 5	1	3	2	4	5	5
Equipo 6	0	4	4	5	4	3
Equipo 7	0	3	2	-1	4	5

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

Por último, las puntuaciones globales son:

Equipo	Puntuación Final
Equipo 1	15
Equipo 2	11
Equipo 3	13
Equipo 4	7
Equipo 5	20
Equipo 6	20
Equipo 7	13

Tabela 4: Valores de score(i) para $R = 5$.

En este ejemplo, el Equipo 5 sería el vencedor. Tiene la misma puntuación que el Equipo 6 (20), pero un número mayor de mejores soluciones conocidas (2 contra 1).

En cualquier etapa del Desafío SBPO 2025, el valor de R será definido como el número de Equipos participantes en la respectiva etapa.

2.5 Limitaciones de computación y de software

Para garantizar una comparación justa de los métodos desarrollados por los diferentes Equipos participantes, cada Equipo deberá presentar su código fuente en todas las etapas. Este programa debe aceptar una instancia de problema de uno de los conjuntos de datos, siguiendo el formato especificado en Sección 3.1. Debe producir un archivo de solución de salida de acuerdo con los estándares definidos en la Sección 3.2. Para cada instancia, los organizadores realizarán *una única* ejecución usando una semilla fija seleccionada aleatoriamente.

Para promover transparencia y justicia, la semilla seleccionada por los organizadores para evaluaciones y clasificaciones de los equipos será divulgada tras la conclusión de cada etapa. Esta semilla será aplicada consistentemente para cada equipo y cada instancia. Es responsabilidad de los participantes garantizar que el valor de la semilla sea pasado como un parámetro al programa ejecutable, posibilitando así la repetibilidad de los experimentos y evaluaciones, especialmente si sus métodos de solución involucran estructuras o componentes probabilísticos. Los programas serán evaluados solo *una única vez* por instancia; *ninguna evaluación adicional será conducida.* En caso de que ocurra variabilidad, los organizadores no pueden ser responsabilizados por cualquier efecto adverso resultante de esa única ejecución.

La computadora que se utilizará para evaluar los programas de los candidatos es una máquina con sistema operativo Linux, 8 vCPUs de 3.3GHz y 16GB de RAM. *El uso de GPUs no estará disponible.*

Los equipos que deseen utilizar un resolvedor de Programación Entera (ILP) deben utilizar el CPLEX 22.11 o el OR-TOOLS 4.11. Los códigos deben ser escritos en JAVA. Ningún otro lenguaje será aceptado.

3 Formato y descripción de los archivos

Esta sección describe el formato de los archivos utilizados en el desafío. Para una mejor comprensión del material presentado aquí, se recomienda leer primero el *archivo de descripción del problema*, que está disponible en este repositorio.

3.1 Entrada

La entrada será proporcionada en un archivo de texto con el siguiente formato:

- Primera línea: contiene tres enteros o, i y a, que representan, respectivamente, el número de órdenes, ítems y corredores.
- Próximas o líneas: cada línea comienza con un entero k, seguido por k pares de enteros, representando el número del ítem y el número de unidades solicitadas por la orden. Dentro de este grupo de líneas y, para j en $\{1, \ldots, o\}$, la j-ésima línea contiene datos relacionados con la orden indexada por $j-1$, significando que las órdenes están numeradas de 0 a $o-1$. Además, se asume que los ítems están indexados de 0 a $i-1$.
- Próximas a líneas: cada línea comienza con un entero l, seguido por l pares de enteros, representando el número del ítem y el número de unidades disponibles en el corredor. Similar al caso de las órdenes, dentro de este grupo de líneas y, para j en $\{1, \ldots, a\}$, la j-ésima línea contiene datos relacionados con el corredor indexado por $j-1$, significando que los corredores están numerados de 0 a $a-1$.
- Última línea: contiene dos enteros, LB y UB, que son los límites inferior y superior del tamaño de la *wave* (un subconjunto de órdenes), respectivamente. Estos límites son expresados en cantidad de unidades de ítems.

Así, la entrada para la instancia descrita en la *Sección de Ejemplos* del archivo de descripción del problema sería:

```
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 Salida

La salida debe ser un archivo de texto con el siguiente formato:

- Primera línea: contiene un entero n, el número de órdenes en la *wave* (subconjunto de órdenes) devuelto por el algoritmo.
- Próximas n líneas: cada línea contiene un entero representando el índice de una orden de la *wave* computada.
- Línea siguiente: contiene un entero m que representa el número de corredores visitados en la *wave* computada.
- Próximas m líneas: cada línea contiene un entero representando el índice de un corredor visitado por la *wave* computada.

Por ejemplo, la salida para la última solución del ejemplo descrito en Sección 3 del archivo de descripción del problema sería:

```
4
0
1
2
4
2
1
3
```

4 Requisitos de código

Como se mencionó en la Sección 2.5, el código debe ser escrito en JAVA, específicamente en la versión 17. Para facilitar el proceso de desarrollo y evaluación, proporcionamos una *plantilla* para el proyecto a ser desarrollado en GITHUB que puede ser clonado de este repositorio.

Una vez que el proyecto haya sido clonado a su repositorio local, la raíz del directorio, llamada `ChallengeSBPO2025` (**¡no la alteres!**), contendrá los siguientes elementos:

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

El archivo `checker.py` es un script PYTHON que verifica si una solución es factible (vea la Sección 4.3), mientras que el script `run_challenge.py` puede compilar y ejecutar fácilmente un código JAVA, en conformidad con las especificaciones de este documento, sobre un conjunto de instancias. Como MAVEN se utilizará para construir el proyecto, el `pom.xml` (Project Object Model) también se proporciona. Contiene información sobre el proyecto y detalles de configuración utilizados por MAVEN. **No altere este archivo, excepto si** CPLEX **o**

OR-TOOLS **son requeridos por el código desarrollado para el desafío** (vea las Secciones 4.1 y 4.2 más abajo). El directorio `src` es donde deben residir los códigos JAVA.

Después de clonar el proyecto disponible en GITHUB, siguiendo la ruta `/src/main/java/org/sbpo2025/challenge/`, se encontrarán tres archivos:

1. `Challenge.java`. Este es el programa *main* del proyecto y define la clase `Challenge` que contiene métodos para leer el archivo de entrada y escribir el archivo de salida.

2. `ChallengeSolver.java`. Este archivo contiene un código incompleto **que debe ser completado por los Equipos para implementar los algoritmos propuestos**. Define la clase `ChallengeSolver`, que es llamada por `Challenge.java` e incluye algunas funciones auxiliares para verificar la factibilidad y calcular el valor objetivo de una solución candidata.

3. `ChallengeSolution.java`. Este archivo define una estructura básica que representa una solución para el problema.

Aunque es altamente recomendado, los equipos no están obligados a mantener el código en `Challenge.java` sin cambios. Sin embargo, independientemente de cualquier modificación realizada, **el archivo ejecutable resultante debe aceptar dos argumentos: las rutas de los archivos de entrada y salida, en ese orden**.

El proyecto se puede compilar usando el comando MAVEN emitido a partir del directorio raíz:

```
mvn clean package
```

Esto generará un archivo JAR, llamado `ChallengeSBPO2025-1.0.jar`, en el directorio `target`.

Para ejecutar el archivo JAR, el siguiente comando puede ser emitido desde el directorio raíz del proyecto:

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

donde `<input-file>` es la ruta para el archivo de entrada a ser leído, y `<output-file>` es la ruta para el archivo de salida a ser escrito.

4.1 Usando CPLEX

Si CPLEX es utilizado por el algoritmo propuesto, todas las líneas del *bloque de dependencias* del archivo `pom.xml` deben ser descomentadas. Además, antes de compilar el código, las dependencias de CPLEX deben ser añadidas al repositorio MAVEN usando el siguiente comando:

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

donde `<cplex-jar-file>` es la ruta al archivo CPLEX JAR (por ejemplo, `/Applications/CPLEX_Studio2211/cplex/lib/cplex.jar`). Observe que **la versión 22.11** del resolvedor será utilizada para la evaluación.

Después de eso, para ejecutar el archivo de aplicación JAR con el CPLEX, debe proporcionar la ruta a la biblioteca de CPLEX, usando el siguiente comando:

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

donde `<cplex-library-path>` es la ruta a la biblioteca de CPLEX en su instalación local (por ejemplo, `/Applications/CPLEX_Studio2211/opl/bin/arm64_osx/`).

4.2 Usando OR-TOOLS

Si decide usar OR-TOOLS, primero debe instalar OR-TOOLS para Java, siguiendo las instrucciones en esta página. Observe que **la versión 9.11** de la biblioteca será utilizada para evaluación. Luego, debe descomentar la respectiva dependencia en el archivo `pom.xml`.

Finalmente, puede ejecutar el archivo JAR con OR-TOOLS usando el siguiente comando:

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

donde `<ortools-library-path>` es la ruta a la biblioteca OR-TOOLS (por ejemplo, `$HOME/Documents/or-tools/build/lib/`).

4.3 Verificador de solución

Un script PYTHON, llamado `checker.py`, es proporcionado para verificar la factibilidad de una solución basado en los archivos de entrada y salida pasados como parámetros, además de calcular el valor objetivo. El script puede ser ejecutado usando el siguiente comando:

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

4.4 Programa PYTHON para la ejecución del desafío

El script `run_challenge.py` puede ser utilizado para compilar y ejecutar el código JAVA para un conjunto de instancias. Requiere que la ruta del código fuente (que contiene el archivo `pom.xml`), el directorio de entrada (instancias) y el directorio de salida (para almacenar las soluciones) sean pasados como parámetros, como se muestra a continuación:

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

Para ejecutar este script, necesitarás el comando `timeout` (o `gtimeout` en macOS) instalado. Puedes instalarlo usando el comando `apt-get install coreutils` (o equivalente) en Linux o `brew install coreutils` en macOS. El script permite que el usuario configure las rutas para las bibliotecas de CPLEX y de OR-TOOLS, además de interrumpir automáticamente la ejecución del programa si el límite de tiempo de 10 minutos es excedido.

4.5 Tiempo límite de ejecución

Recuerde que se asignará un límite de tiempo de 10 minutos para computar una solución para cada instancia. Por lo tanto, se recomienda usar algún paquete JAVA, como `StopWatch`, para monitorear el tiempo de ejecución y garantizar que se escriba un archivo de salida *dentro de este límite de tiempo*.

5 Discusión y actualizaciones

Para ayudar a los Equipos a gestionar sus proyectos, los organizadores se comprometen a comunicar cualquier cambio que pueda ocurrir durante el desafío con la máxima brevedad. Esto incluye aclaraciones sobre la descripción del problema o cambios en las instancias del problema, por ejemplo. Estos cambios solo se realizarán si son estrictamente necesarios y serán comunicados a través del GITHUB del repositorio del desafío. Los participantes pueden contactar a los organizadores a través del correo electrónico sor-challenges@mercadolibre.com, pero solo en caso de dudas o problemas significativos.

6 Propiedad intelectual

Los equipos participantes del desafío mantendrán los derechos de autor sobre los programas de computadora desarrollados durante el desafío. MERCADO LIVRE y cualquier tercero pueden usar la información proporcionada por los participantes a través de informes técnicos, artículos científicos y presentaciones orales, pero no podrán usar un programa de computadora sin el acuerdo del equipo que desarrolló dicho programa.

Los participantes del desafío no pueden afirmar que tienen una asociación con MERCADO LIVRE o cualquier contrato con MERCADO LIVRE o con entidades del grupo MERCADO LIVRE, incluso si ganan el desafío. Solo pueden alegar que participaron (o que fueron clasificados o ganadores) del desafío, si es el caso.

7 Condiciones generales

Todas las decisiones sobre el Desafío, incluyendo aclaraciones sobre cualquier duda acerca de la mecánica de este proyecto, quedan bajo responsabilidad de MERCADO LIVRE.

Los Participantes reconocen y acuerdan expresamente que el Desafío de excelencia **no** establece ningún vínculo laboral ni remuneración salarial.

MERCADO LIVRE se reserva el derecho de modificar el presente Reglamento en cualquier momento, total o parcialmente, mediante previo aviso a los Participantes, a través de un comunicado por escrito, siendo cierto que la continuidad del Participante en el Desafío, tras tales modificaciones, se presumirá como aceptación tácita de las nuevas reglas.

La participación en el Desafío no generará al Participante ningún otro derecho o ventaja que no esté expresamente previsto en este Reglamento.

Los Participantes, al momento de su registro, autorizan a MERCADO LIVRE a utilizar, para fines administrativos y de marketing, todos sus datos registrales y la información de consumo generada, siempre que se respeten las disposiciones legales vigentes.

Las dudas que existan en este Reglamento serán tratadas directamente entre el participante y MERCADO LIVRE y podrán ser enviadas al correo electrónico de contacto: `sor-challenges@mercadolibre.com`.

El Desafío será regulado exclusivamente por este Reglamento, que puede ser sustituido o alterado a criterio único y exclusivo de MERCADO LIVRE. En caso de pérdida de credenciales de un integrante del Equipo de la institución de estudio indicada en el registro, el Equipo será descalificado.

En caso de verificación de fraude, intento o sospecha de fraude de un Equipo, la puntuación atribuida al Equipo será anulada. Una vez confirmado el fraude, el Equipo será automáticamente excluido del Desafío, sin perjuicio de las medidas legales aplicables.

MERCADO LIVRE se reserva el derecho de decidir sobre cualquier punto no especificado en este Reglamento, siendo su decisión soberana e inapelable.

Los casos omisos y/o no previstos en este Reglamento serán de buena fe, resueltos por la comisión de MERCADO LIVRE responsable del desafío.

Los datos recogidos en el Desafío son utilizados para regular la gestión del Desafío y acciones que se realicen en él, identificación de los Usuarios, comunicación, seguimiento del desempeño, disponibilidad de extractos, elaboración de informes y demás análisis aplicables, inclusive para fines estadísticos.

MERCADO LIVRE se compromete a respetar la confidencialidad de todos y cualquier dato e información relativa a una persona física que pueda ser identificada o identificable ("Datos Personales") o comerciales que sean recolectados a través de internet, independientemente del sistema de recolección utilizado, sean de los Participantes o de cualquier tercero (los "Datos Electrónicos"), observando, para tal fin, la Ley nº 13.709/2018 (Ley General de Protección de Datos Personales), la Ley nº 12.965/2014 (Marco Civil de Internet) y el Decreto nº 8.771/2016 (Reglamento del Marco Civil de Internet), así como cualquier otra ley relacionada a la protección de datos personales obtenidos a través de

internet que se promulgue en la República Federativa de Brasil o que entre en vigor durante la vigencia de este Contrato.

El uso de la información personal de los participantes del Desafío como nombre, CPF, correo electrónico y whatsapp, información que sirve para hacer identificables a todos los que participan del Desafío durante su vigencia, se realizará de acuerdo con la legislación, especialmente la Ley nº 13.709/18, ya que los procesos de tratamiento solo podrán ser realizados para propósitos legítimos, específicos, explícitos y previamente informados a los titulares de los Datos Personales.

El participante, al aceptar este reglamento, acepta el tratamiento de sus datos personales por parte de MERCADO LIVRE.

MERCADO LIVRE garantiza que adoptará medidas de prevención para todos los datos personales de los participantes registrados en la campaña.

Todos los datos personales de los participantes serán usados única y exclusivamente en virtud del Desafío, estando, por lo tanto, prohibido el uso y/o tratamiento de esos Datos Personales sin el previo consentimiento de sus titulares, para finalidades diferentes de las expresamente determinadas en este reglamento.

MERCADO LIVRE y el participante acuerdan que, tras el término de la vigencia de la campaña, todos los Datos Personales registrados serán eliminados.

Se elige el Foro de la Comarca de San Paulo/SP para cualquier disputa judicial que, eventualmente, pudiera existir en virtud de las disposiciones previstas en este Reglamento.