

Manual de usuario de la herramienta *IGD-IRACE*

1. ¿Cómo obtengo la herramienta?

Para obtener la herramienta, diríjase a el repositorio de GitHub y descarga el *script* llamado `IGD_IRACE.py`. En caso de necesitar un ejemplo, clona el repositorio para respetar la estructura del repositorio.

2. ¿Qué requisitos necesito para usar la herramienta?

La ejecución de la herramienta necesita obligatoriamente que la computadora tenga instalado el sistema operativo Linux. La herramienta *IGD-IRACE* fue probada en máquinas con las distros Ubuntu y Linux Mint. Lo anterior deja ver que la correcta ejecución en la herramienta no está asegurada en otras distros, a reserva de aquellas que tengan como base Debian.

La herramienta hace uso de la librería PyQT5, misma que es descargada al ejecutar el *script* `IGD_IRACE.py` en caso de que dicha librería no se encuentre instalada. Debido a que IRACE es una librería del lenguaje de programación R, la computadora debe de contar con dicho lenguaje. R se obtiene en la siguiente liga, en esta encontrará como instalarlo. Tras instalar R en la computadora, debe abrir una terminal y escribir la letra R. Seguidamente deberá presionar ENTER y en la terminal lucirá como en la Figura 1. Para instalar la librería IRACE, deberá escribir en la sesión de R la siguiente instrucción:

```
1 install.packages("irace")
```

De instalar correctamente la librería, la terminal deberá mostrar que se ha instalado de manera correcta, tal como lo muestra la Figura 2. Tras instalar R y IRACE, deberá modificar el archivo `.bashrc` ubicado en la carpeta raíz del sistema operativo. Esto se deberá realizar de la siguiente manera:

1. Teniendo la sesión abierta de R en la terminal, deberá escribir en la misma lo siguiente:

```
1 system.file(package = "irace")
```

Listing 1: Ruta para añadir en el `.bashrc`

La ruta que aparece tras ejecutar dicha instrucción, será importante en el siguiente paso.

2. Abra y modifique el `.bashrc` añadiendo las variables de entorno. Para abrir el archivo `.bashrc` deberá escribir en la terminal de Linux el siguiente comando:

```
1 gedit .bashrc
```

Listing 2: Como abrir el `.bashrc`

Gedit es el editor de texto de Ubuntu, use el que tenga instalado en su computadora. Modifique el `.bashrc`, escribiendo las siguientes instrucciones al final del archivo:

```
1 export IRACE_HOME="/home/user/.../X86_64-pc-linux-gnu-library/4.1/irace"  
2 export PATH=${IRACE_HOME}/bin/:$PATH
```

Listing 3: Lineas a escribir en el `.bashrc`

Note que la ruta que se debe de asignar en la variable de entorno `IRACE_HOME` es la misma que se obtiene de escribir en la sesión abierta de R, `system.file(package = "irace")`

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
~$ R

R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

[Previously saved workspace restored]

> █
```

Figura 1: Verificación de la instalación de R en terminal.

3. Previo al uso de IGD-IRACE

Para usar la herramienta, deberá colocar el *script* IGD_IRACE.py en la carpeta en la que se encuentra su código de python. Por ejemplo, el archivo Template.py de la Figura 3 es el algoritmo que se utilizará para resolver el problema de optimización. El archivo Template.py deberá tener como *output* en la terminal un número, que representa el mejor valor encontrado por el algoritmo en la última iteración.

Es necesario que la siguiente expresión sea agregada al *script* Template.py en la primer línea:

```
1 #!/usr/bin/env python3
```

De igual manera, será importante que los parámetros del algoritmo sean ofrecidos al interprete de python a través de la línea de comandos. Esto se realiza importando la librería `sys` al *script* de python. Como ejemplo de esto, considere que el algoritmo Template.py requiere de los siguientes parámetros:

- Número de iteraciones
- Número de individuos
- Probabilidad de mutación
- Probabilidad de cruce

```
1 #!/usr/bin/env python3
2
3 import sys
4 path          = sys.argv[1]
5 n_iteraciones = int(sys.argv[2])
6 n_individuos  = int(sys.argv[3])
7 prob_mutacion = float(sys.argv[4])
8 prob_cruce    = float(sys.argv[5])
9 semilla       = int(sys.argv[6])
```

Listing 4: Ejemplo del archivo Template.py

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda

Installing package into '/home/          /R/x86_64-pc-linux-gnu-library/4.1'
(as 'lib' is unspecified)
probando la URL 'https://cloud.r-project.org/src/contrib/irace_3.5.tar.gz'
Content type 'application/x-gzip' length 1243337 bytes (1.2 MB)
=====
downloaded 1.2 MB

* installing *source* package 'irace' ...
** package 'irace' successfully unpacked and MD5 sums checked
** using staged installation
** libs
gcc -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -ffile-prefix-map=/build/r-base-4A2Reg/r-base-4.1.2=.
-fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g -c iracebin/irace
.c -o iracebin/irace.o
gcc -g -O2 -ffile-prefix-map=/build/r-base-4A2Reg/r-base-4.1.2=. -fstack-protector-strong -Wformat -Werror=for
mat-security -Wdate-time -D_FORTIFY_SOURCE=2 -g iracebin/irace.o -o iracebin/irace
gcc -I"/usr/share/R/include" -DNDEBUG      -fpic  -g -O2 -ffile-prefix-map=/build/r-base-4A2Reg/r-base-4.1.2=.
-fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FORTIFY_SOURCE=2 -g -c iracebin/ablat
ion.c -o iracebin/ablation.o
gcc -g -O2 -ffile-prefix-map=/build/r-base-4A2Reg/r-base-4.1.2=. -fstack-protector-strong -Wformat -Werror=for
mat-security -Wdate-time -D_FORTIFY_SOURCE=2 -g iracebin/ablation.o -o iracebin/ablation
installing via 'install.libs.R' to /home/aaronlp/R/x86_64-pc-linux-gnu-library/4.1/00LOCK-irace/00new/irace
** R
** inst
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded from temporary location
** testing if installed package can be loaded from final location
** testing if installed package keeps a record of temporary installation path
* DONE (irace)

The downloaded source packages are in
  '/tmp/Rtmp3Pnqnk/downloaded_packages'
>

```

Figura 2: Instalación de IRACE en la terminal

La variable `path` definida en la línea 4 del Listing 4 está relacionada con la instancia que se quiere probar, misma que se encuentra dentro de la carpeta *Instancias* de la Figura 3. Seguidamente, las variables definidas en las líneas 5, 6, 7 y 8 del mismo bloque, se refiere a los parámetros mencionados con anterioridad. Note que dichos parámetros son convertidos a su tipo de dato, para evitar que el código no produzca errores.

En la práctica, IRACE mandará a llamar al interprete de python de la siguiente manera:

```
1 python3 /ruta/a/instancia p_iteraciones p_individuos p_mutacion p_cruce semilla
```

Listing 5: Como IRACE y R mandan a llamar al interprete de python

En ese sentido, `p_iteraciones`, `p_individuos`, `p_mutacion` y `p_cruce` son números que R irá asignando de acuerdo a los intervalos que el usuario defina. Si usted necesitara añadir o eliminar parámetros, deberá ajustar dicho bloque de variables del código de acuerdo a sus necesidades.

4. Ejecutar IGD-IRACE

Una vez que se ha completado el ajuste de la plantilla *Template.py*, usted estará listo para usar la herramienta. Para abrir IGD-IRACE, deberá abrir una terminal en la carpeta donde se encuentre la plantilla y el *script* IGD_IRACE.py. En la figura 4 se muestra como debe ser realizado esto. Note que debe escribirse:

```
1 python3 IGD_IRACE.py
```

Listing 6: Como ejecutar la interfaz IGD-IRACE

Por favor no cierre la terminal ni la herramienta hasta que en la terminal aparezca una leyenda que indique que el proceso de optimización de los parámetros ha finalizado. De la misma manera, se le pide que la ruta en la que se encuentran los scripts *Template.py* y *IGD_IRACE.py* no contenga espacios como parte del nombre de las carpetas. Esto se pide, debido a que el manejo de espacios en la terminal y en documentos en los que se insertan rutas, son sensibles a los espacios, el uso de mayúsculas y minúsculas.

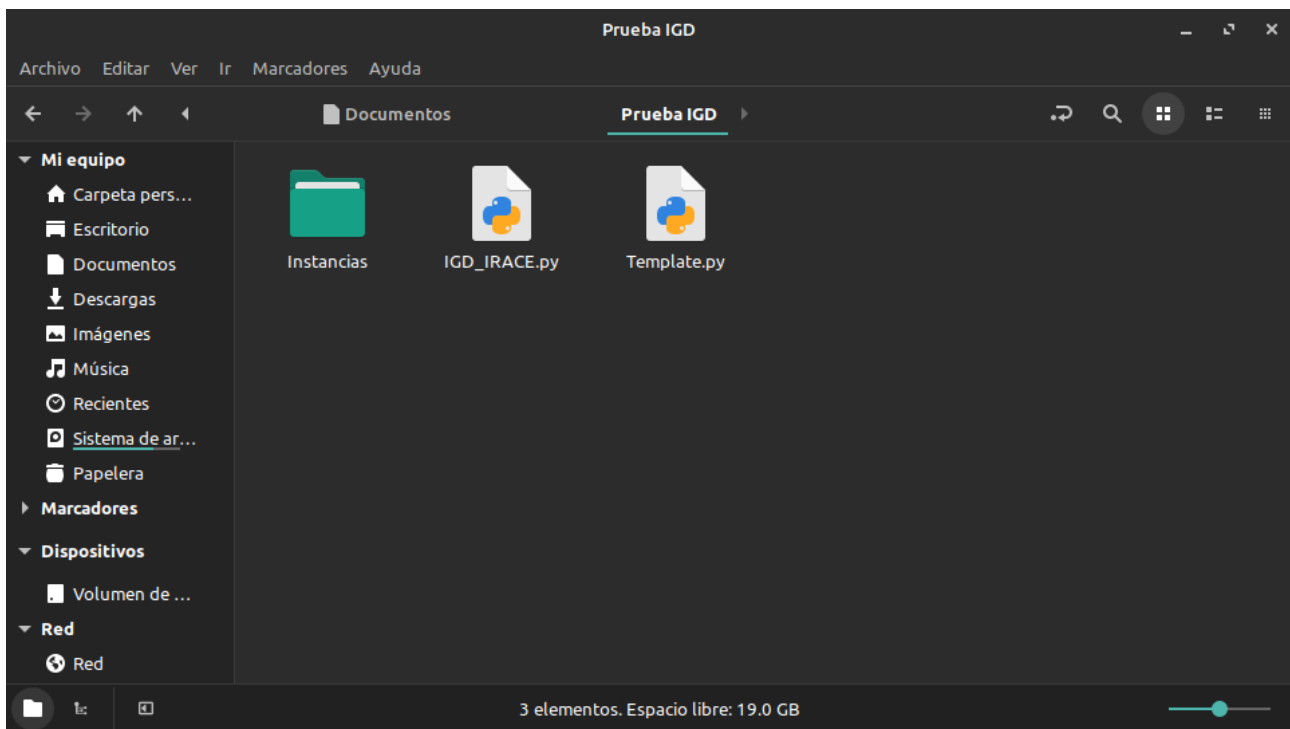


Figura 3: Carpeta de ejemplo. El archivo IGD-IRACE debe de ser colocado dentro de este directorio.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
~/Documentos/  /Prueba_IGD$ ls -l
total 32
-rw-rw-r-- 1          21668 abr 25 23:41 IGD_IRACE.py
drwx----- 5          4096 may 27 2023 Instancias
-rw-rw-r-- 1          3600 jun  9 2023 Template.py
~/Documentos/  /Prueba_IGD$ python3 IGD_IRACE.py
```

Figura 4: Como debe de abrirse IGD-IRACE

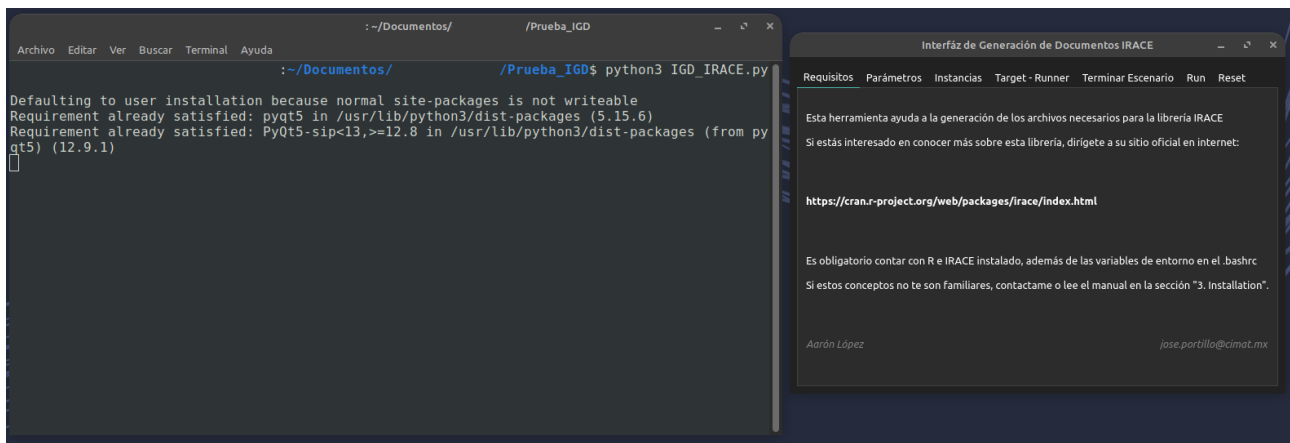


Figura 5: Apertura de la herramienta IGD-IRACE.

Tras ejecutar el *script* que inicia la interfaz gráfica, se descargará (en caso de no contar con ella) la librería PyQT5. El tiempo de espera hasta la aparición de la interfaz dependerá de la velocidad de su red y si la librería se encuentra instalada. Tras la instalación de la paquetería requerida, se desplegará una interfaz similar a la mostrada en la Figura 5. En la izquierda se muestra la terminal que mostrará mensajes de guía. A la derecha se muestra la herramienta IGD-IRACE. El orden de usar las pestañas es de izquierda a derecha:

Requisitos >> Parámetros >> Instancias >> Target - Runner >> Terminar Escenario >> Run.

En caso de que se cometa un error en el proceso de ingresar los parámetros y campos solicitados, deberá dirigirse a la pestaña *Reset* y hacer clic en el botón *Reseteo Archivos*. Esto tendrá como consecuencia que los archivos generados, serán eliminados, excluyendo el *Template.py* y *IGD_IRACE.py*. Al ser eliminados, se reiniciará la herramienta y tendrá que reiniciar el proceso de ingresar los campos requeridos. Por esta razón, es necesario que realice el proceso de ingresar los datos de manera cautelosa.

4.1. Requisitos

En dicha pestaña se realizará una breve explicación de los requisitos necesarios. Si usted ha seguido paso a paso esta guía, habrá cumplido con los requisitos y podrá seguir a la siguiente pestaña.

4.2. Parámetros

En esta pestaña, encontrará distintos campos que necesitará llenar. En la Figura 6 se muestra un ejemplo del contenido de dicha pestaña. El primer paso será hacer click en el botón **Seleccionar template**, en donde desplegará un menú que le permitirá seleccionar la plantilla *Template.py*. Al hacer esto, se le concederán los permisos para hacer de este *script* un ejecutable, por lo que usted no tendrá que hacerlo manualmente. Seguidamente, deberá hacer click en el botón **Generar archivos de IRACE**, lo que generará los archivos que se irán escribiendo conforme a las necesidades que usted tenga.

Seguidamente, deberá de escribir el **Número de parámetros del tipo entero y continuo**. Retomando el ejemplo mostrado con anterioridad, el algoritmo mostrado necesita de los parámetros número de iteraciones, individuos, probabilidad de mutación y probabilidad de cruce. El primer par de estos parámetros son del tipo entero, por lo que deberá poner en la última columna la letra "i" para que IRACE entienda que dicho parámetro puede tomar valores entre el mínimo y máximo que deberá ingresar en las columnas correspondientes. Los parámetros probabilidad de mutación y cruce son del tipo continuos, por lo que deberá ingresar la letra "r" en la cuarta columna. De igual manera, deberá ingresar los números que definirán el rango de posibles valores que dichos parámetros pueden tomar.

Si usted desea agregar un parámetro que tome valores de un conjunto discreto de números, deberá ingresar en el espacio **Número de parámetros de tipo discreto** el número de variables requeridas. De igual manera, encontrará una tabla en la que deberá colocar el nombre de la variable y seguidamente deberá ingresar los valores que dicha variable puede tomar. Cabe resaltar que dichos valores deberán estar separados por una coma.

Figura 6: Pestaña Parámetros

Encarecidamente le solicito que no deje filas ni columnas sin contestar; si requiere únicamente de dos parámetros, ingrese el número entero que se adapta a su necesidad. Tras ingresar los parámetros necesarios, deberá hacer click en el botón **Añadir en parameters.txt**

5. Instancias

En esta pestaña deberá hacer click en el botón **Seleccionar carpeta de instancias**. Dicho botón desplegará una pestaña en la que deberá seleccionar la carpeta que contiene todas las instancias que usted requiere como conjunto de entrenamiento. Seguidamente, deberá ingresar en el espacio **Número de instancias** el número de instancias que se utilizarán para el entrenamiento. Notará que se desplegará una tabla que permitirá ingresar el nombre de las instancias que se alojan en la carpeta seleccionada. Encarecidamente le solicito que el nombre de las instancias no contengan espacios y que el nombre sea correcto. No escribir las instancias de manera correcta, generará errores que serán difícil de detectar en la terminal. Tras ingresar los nombres de dichas instancias, deberá hacer click en los botones **Escribir en instances-list.txt** y **Escribir en escenario.txt**. Esto actualizará el contenido de dichos archivos alojados en la carpeta. Para ver dicha pestaña, diríjase a la Figura 7.

5.1. Target- Runner

En dicha pestaña (Figura 8) se muestran un conjunto de instrucciones que deberá realizar a mano. En primer lugar deberá de modificar las líneas 37 y 39 del archivo `target-runner` con su editor de texto favorito. Reemplace las instrucciones existentes por las que se muestran en el Listing 7.

```
1 EXE="./Template.py"
2 EXE_PARAMS="$INSTANCE ${CONFIG_PARAMS} $SEED"
```

Listing 7: Instrucciones a reemplazar en target-runner



Figura 7: Pestaña de instancias

Usted podría cambiar el nombre de su plantilla por el de su preferencia, pero considere que en tal caso, debe cambiar dicho nombre en el `target-runner`. Si usted ha seguido las instrucciones de este manual, verá que el resto de indicaciones ya las habrá realizado.

5.2. Terminar Escenario

En esta pestaña (Figura 9), deberá ingresar el número de experimentos que deberá de realizar a lo largo de la optimización de los parámetros. Ingrese un número mayor a las 1000 repeticiones. Seguidamente deberá ingresar el número 1 en el espacio destinado a la paralelización de la optimización. Esto ayudará a distribuir el trabajo en los distintos núcleos del procesador y terminar la tarea en menor tiempo. Finalmente deberá introducir el número de dígitos que la librería retornará por cada parámetro a optimizar. Tras completar dichos valores, deberá hacer click en el botón **Finalizar escenario.txt** para terminar de configurar los archivos de entrada que la librería IRACE necesita.

5.3. Run

Para iniciar la optimización deberá hacer click en el botón **Optimizar!**. El proceso de optimización dependerá de los parámetros a optimizar, el número de dígitos, el número de máximos experimentos, las instancias introducidas y lo optimizado que se encuentre el archivo `Template.py`. Una vez que la optimización haya terminado, en la terminal saldrá un mensaje que indica que es seguro cerrar la herramienta IGD-IRACE y la terminal. Los resultados de dicho proceso se encontrarán en un archivo llamado **Salida_optimización.txt**, en el que deberá leer la parte final del mismo para encontrar los conjuntos de parámetros que deberán probarse a fin de determinar el mejor de ellos.

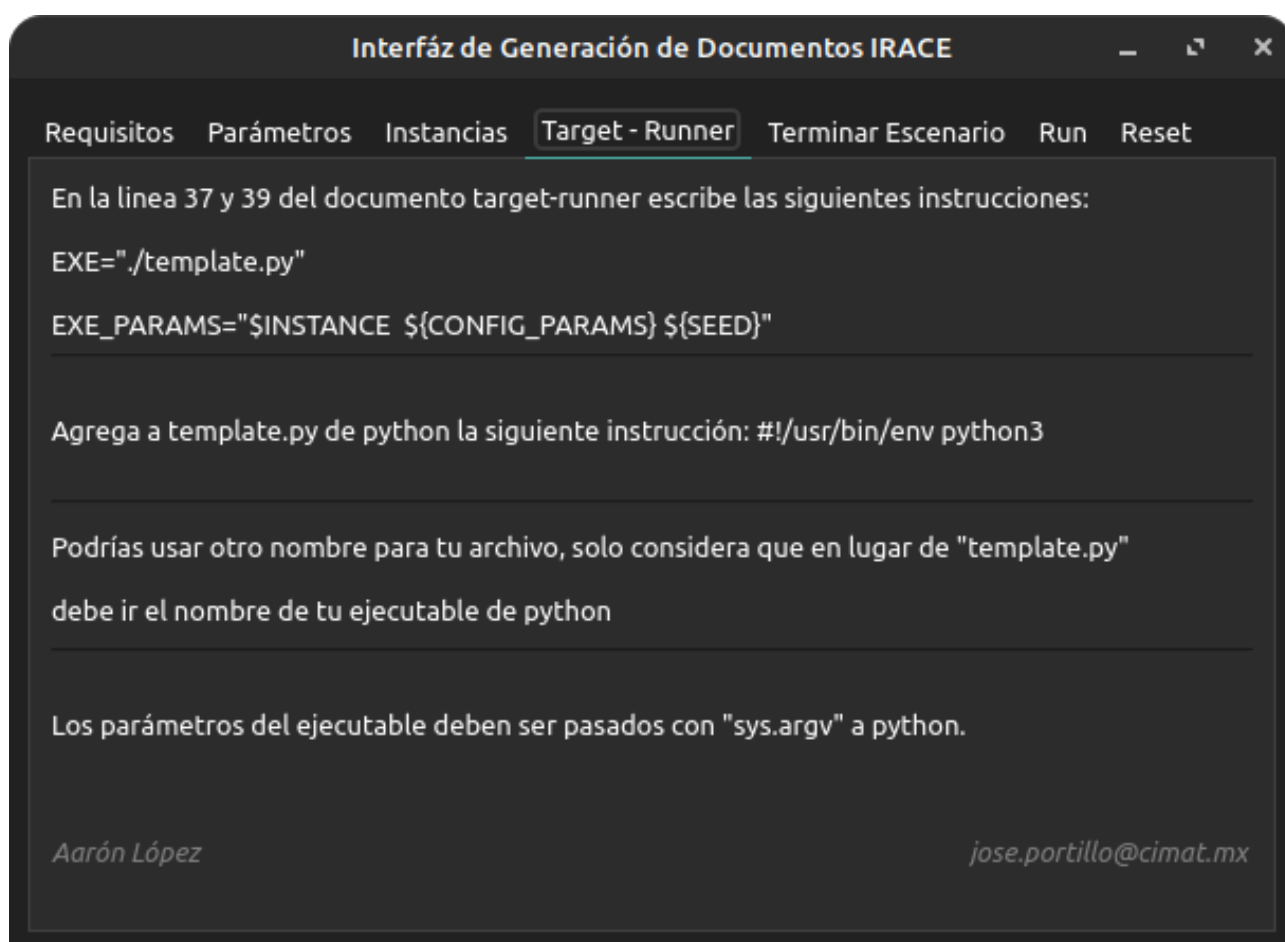


Figura 8: Pestaña Target-Runner

Interfáz de Generación de Documentos IRACE

Requisitos Parámetros Instancias Target - Runner Terminar Escenario Run Reset

Número máximo de experimentos:

Realizar el proceso en paralelo (Si = 1, No = 0):

Cantidad de decimales:

Finalizar escenario.txt

Aarón López jose.portillo@cimat.mx

Figura 9: Pestaña Terminar Escenario

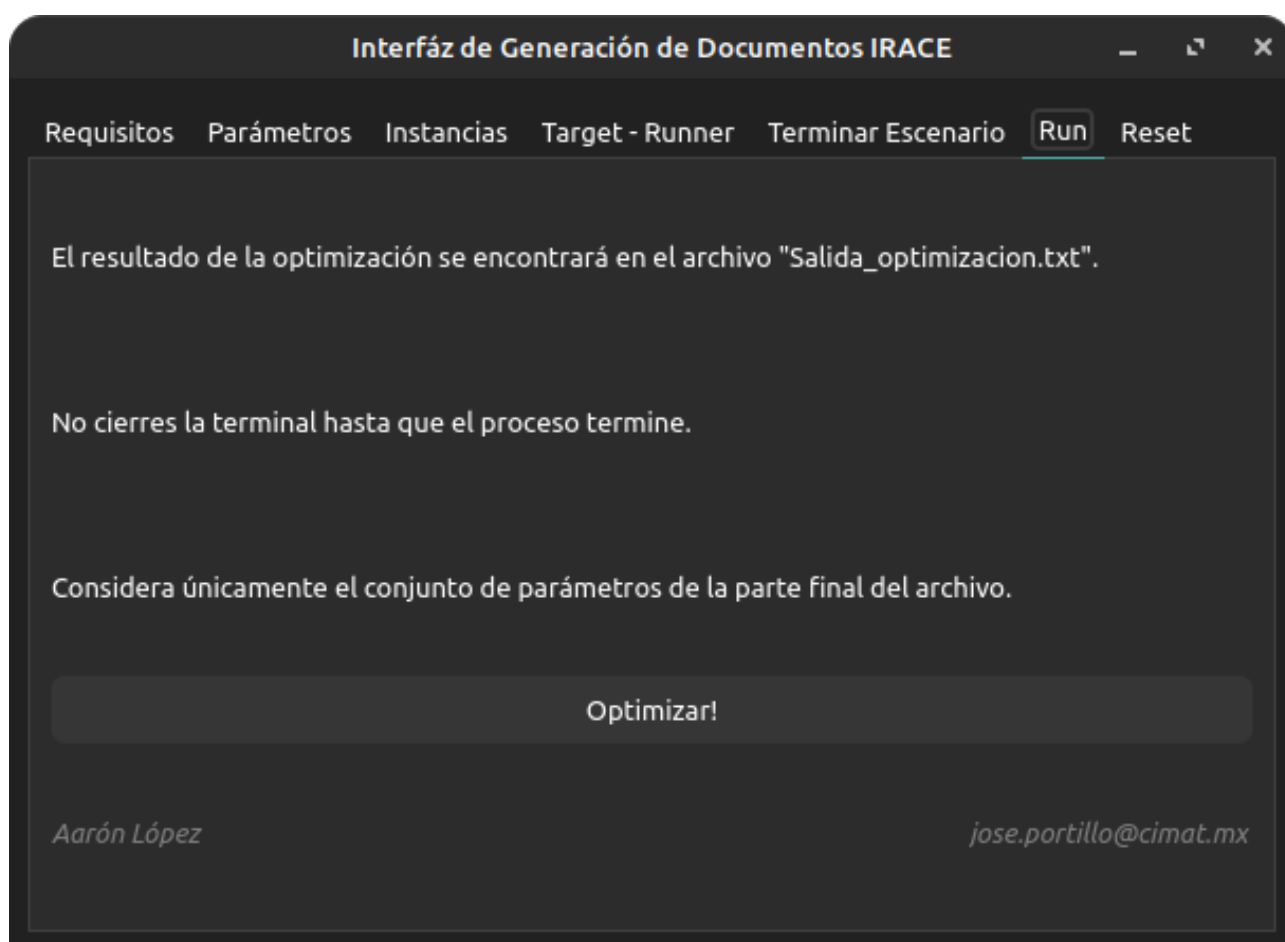


Figura 10: Pestaña Run