

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA INFORMATICA

**INTERFAZ DE USUARIO PARA LA CONFIGURACIÓN AUTOMÁTICA DE
ALGORITMOS: CONTROL DE EJECUCIÓN**

JOSÉ PABLO ARANCIBIA LINKER

INFORME FINAL DE PROYECTO PARA
OPTAR AL TITULO PROFESIONAL DE
INGENIERO EJECUCIÓN INFORMATICA

ENERO, 2020

Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

**INTERFAZ DE USUARIO PARA LA CONFIGURACION
AUTOMATICA DE ALGORITMOS: CONTROL DE EJECUCIÓN**

JOSÉ PABLO ARANCIBIA LINKER

Profesor guía: **Leslie Pérez Cáceres**

Profesor Co-referente: **Nibaldo Rodríguez Aburto**

Carrera: **Ingeniería de Ejecución en Informática**

ENERO, 2020

Índice

Resumen.....	iii
Abstract.....	iii
Glosario de Términos.....	iv
Lista de Figuras	v
Lista de Tablas.....	vi
1. Introducción.....	1
2. Descripción general de la situación en estudio	2
2.1. Definición de Objetivos.....	2
2.1.1. Objetivo General	2
2.1.2. Objetivos Específicos	2
3. Situación en estudio.....	3
3.1. Descripción de la situación actual.....	3
3.2. Identificación de los problemas detectados.....	3
4. Propuesta Solución.....	4
4.1. Descripción general de la solución.....	4
4.2. Modelo de proceso de desarrollo.....	4
4.3. Herramientas de desarrollo.....	4
5. Gestión del Proyecto.....	5
5.1. Carta Gantt.....	5
5.1.1. Primera fase carta gantt	5
5.1.2. Primera iteración carta gantt.....	5
5.1.3. Segunda iteración carta gantt	6
5.1.4. Tercera iteración carta gantt	6
6. Especificación de requerimientos	7
6.1. Requerimientos funcionales	7
6.2. Mockups	9
6.2.1. Control de Ejecución: Output línea de comandos	9
6.2.2. Control de Ejecución: Resumen	10
6.2.3. Control de Ejecucion: Plots de Performance.....	11
6.2.4. Control de Ejecución: Plots de Frecuencia.....	12
6.3. Aplicación final.....	13
6.3.1. Sección “Summary”	13
6.3.2. Sección “Performance”	14

6.3.3.	Sección “Frequency”	15
6.3.4.	Detalles de la aplicación.....	16
6.4.	Casos de Uso	17
6.5.	Modelo de procesos de negocio	18
7.	Estudio de factibilidad	19
7.1.	Factibilidad Técnica.....	19
7.2.	Factibilidad Económica	20
7.2.1.	Recurso Humano	20
7.2.2.	Hardware	20
7.2.3.	Software	21
7.3.	Factibilidad Operacional	22
7.4.	Factibilidad Legal.....	22
8.	Desarrollo de la aplicación	23
9.	Expectativas a futuro	24
10.	Conclusión.....	25
	Referencias.....	26

Resumen

El siguiente documento describe el proceso de estudio, desarrollo e implementación del Proyecto “Interfaz de usuario para la configuración automática de algoritmos: Modulo Control”. El presente informe contiene la descripción de la situación en estudio, la descripción del problema y la propuesta solución para los problemas detectados. Posteriormente se muestra la planificación del proyecto mediante una carta Gantt y luego se definen los requerimientos y descripciones graficas de cómo funciona y las características del software. Por último, se finaliza el presente informe con la descripción del proceso de desarrollo de la aplicación, expectativas a futuro de la interfaz y las conclusiones finales del proyecto.

Abstract

The following document describes the process of the study, development and, implementation of the project “User Interface for the automatic configuration of algorithms: Control Module”. This report contains a description of the situation under study, the description of the problem and the proposed solution for the detected problems. Subsequently, it shows the scheduling of the project through a Gantt card and then are defined the requirements and the graphic descriptions of how the software works and the characteristics of it. Finally, this report ends with the descriptions of the application development process, future expectations of the interface and the conclusions of the project.

Glosario de Términos

Iterated F-Race: Algoritmo cuya función es buscar la mejor configuración de un conjunto de configuraciones candidatas bajo evaluaciones estocásticas.

IRACE: Paquete de R que implementa el procedimiento iterated Racing, el cual es una extensión del procedimiento Iterated F-Race

Framework: Conjunto estandarizado de conceptos, prácticas y criterios para enfocar una problemática en particular con una estructura de software pre definida.

Shiny: Framework que utiliza plantillas html para la realización de interfaces gráficas y utiliza el lenguaje R para la parte funcional.

Lista de Figuras

Figura 5.1 Carta Gantt primera fase	5
Figura 5.2 Carta Gantt primera iteración.....	5
Figura 5.3 Carta Gantt segunda iteración	6
Figura 5.4 Carta Gantt tercera iteración	6
Figura 6.1 Mockup Command Line	9
Figura 6.2 Mockup Summary	10
Figura 6.3 Mockup Performance Plots	11
Figura 6.4 Mockup Frequency Plots	12
Figura 6.5 Sección Summary Aplicación Final.....	13
Figura 6.6 Sección Summary 2 Aplicación Final	14
Figura 6.7 Sección Performance Aplicación Final	14
Figura 6.8 Sección Frequency Aplicación Final	15
Figura 6.9 Sección Frequency 2 Aplicación Final	15
Figura 6.10 Detalles de la Aplicación SideBar.....	16
Figura 6.11 Detalles de la Aplicación Status.....	16
Figura 6.12 Diagrama caso de uso modulo control de ejecución	17
Figura 6.13 Modelo de procesos de negocio modulo control de ejecución	18

Lista de Tablas

Tabla 6.1 Requerimientos Funcionales	7
Tabla 7.1 Tabla de factibilidad técnica.....	19
Tabla 7.2 Factibilidad económica recurso humano	20
Tabla 7.3 Factibilidad económica hardware.....	20
Tabla 7.4 Factibilidad económica software	21

1. Introducción

Actualmente, los algoritmos de optimización son utilizados tanto en el área académica como en la industria, diseñados para encontrar respuestas o resultados de problemas complejos de manera rápida y efectiva.

Es común que estos algoritmos no posean una interfaz gráfica con la que el usuario pueda interactuar, ya que estos usualmente se desarrollan para un problema específico y son complejos de programar.

En este proyecto en específico se trabajará en base a un algoritmo de optimización que utiliza el procedimiento Iterated Racing, el cual es una extensión del procedimiento Iterated F-Race, implementado como un paquete de R llamado IRACE, para crear una interfaz intuitiva, que facilite la visualización de los resultados que entrega este algoritmo y que tenga la capacidad de refrescar en tiempo real los datos que entregue IRACE.

En el presente informe se dará a conocer el problema que da origen al proyecto junto con su propuesta de solución, el objetivo general y objetivos específicos, la metodología de desarrollo, planificación, estudio de factibilidad y las herramientas que se utilizarán para llevar a cabo el proyecto.

2. Descripción general de la situación en estudio

Actualmente, los algoritmos de optimización requieren la configuración de una cantidad elevada de parámetros para que su rendimiento sea óptimo. A esto se agrega que existe una gran variedad de algoritmos de optimización con variables de control que deben ser establecidas con cuidado, ya que encontrar una configuración correcta, cuando existen cientos de parámetros, a ensayo y error se puede dificultar demasiado, además de requerir una cantidad enorme de tiempo.

2.1. Definición de Objetivos

2.1.1. Objetivo General

Desarrollar una interfaz gráfica para el paquete de R IRACE, que permita visualizar los datos entregados por este durante la ejecución del sistema en tiempo real

2.1.2. Objetivos Específicos

- Analizar y aprender el funcionamiento del software IRACE.
- Aprender sintaxis del lenguaje R.
- Aprender a utilizar las herramientas que posee R para desarrollar la solución.
- Analizar y aprender el funcionamiento del framework Shiny para el desarrollo de la interfaz.
- Diseñar una interfaz gráfica aplicando conceptos de UX.
- Entregar un informe que contenga todas las etapas del desarrollo del proyecto.

3. Situación en estudio

3.1. Descripción de la situación actual

El presente proyecto se centrará específicamente en IRACE (*Iterated Race for Automatic Algorithm Configuration*), un software multiplataforma codificado en “R”, que implementa el método Iterated Race, el cual es una extensión del procedimiento Iterated F-Race, para la configuración de optimización de algoritmos, el cual consiste en ajustar sus parámetros mediante la búsqueda de la configuración más adecuada dado un conjunto de instancias de un problema de optimización.

Actualmente, IRACE es un software utilizada tanto en la industria como en el ámbito académico, esto quiere decir que existe un gran número de usuarios que interactúan con el sistema. Este software consiste en ajustar sus parámetros mediante la búsqueda de la configuración más adecuada dado un conjunto de instancias de un problema de optimización. La pre-configuración para la ejecución de IRACE, para usuarios inexpertos, podría resultar dificultosa para estos, ya que esta acción se realiza mediante líneas de comandos y archivos de texto. Por otra parte, durante la ejecución del software, el usuario no posee un feedback sobre lo que está ocurriendo durante este periodo, por lo que no existe un seguimiento de ejecución. Esto repercute en la experiencia del usuario al utilizar IRACE, por lo tanto, el uso del software se vuelve tedioso y poco amigable

3.2. Identificación de los problemas detectados

Los problemas detectados en el presente proyecto son los siguientes:

- El software IRACE es ejecutado exclusivamente mediante línea de comandos, dificultando su acceso a usuarios menos experimentados en el área de la computación.
- La interfaz actual de IRACE (línea de comandos) no permite una interacción real al usuario, por lo que este no puede visualizar de manera amigable los resultados entregados por el software.
- Dificultad al visualizar los gráficos que irace entrega ya que el output es mostrado por líneas de comando y por este medio es imposible mostrar una representación gráfica compleja como son los gráficos que entrega irace luego de la ejecución

4. Propuesta Solución

4.1. Descripción general de la solución

Se busca desarrollar una interfaz gráfica para toda la fase de ejecución del algoritmo, que contempla: Set-up de escenarios, Control de ejecución y Generación de reportes. Esto con la finalidad de disminuir la complejidad de uso de IRACE.

Este proyecto, se enfocará específicamente en la fase de Control de ejecución. Se desarrollará una interfaz que permita al usuario visualizar todo lo que está ocurriendo durante la ejecución de IRACE.

Las ventajas que se obtendrán con la interfaz serán que el usuario estará informado de todos los resultados que el software arroje, podrá visualizar plots para realizar comparaciones con resultados y parámetro ingresados.

Un sistema, al poseer una interfaz, obtiene beneficios como que la interacción con el sistema se vuelve más amena y simplificar el trabajo del usuario al tener la información en una sola ventana, por otro lado, el sistema pasa a ser más intuitivo, ya que la interfaz contiene botones y secciones con nombres representativos y permite que el usuario sepa en cada momento a que área o modulo está accediendo.

4.2. Modelo de proceso de desarrollo

El modelo de proceso de desarrollo a utilizar será un modelo iterativo, ya que el presente proyecto estará basado en entrevistas semanales con el cliente para la modificación o recolección de nuevos requisitos para el sistema. Este modelo permite que exista una retroalimentación entre el desarrollador y el cliente permitiendo que en el proceso de implementación y desarrollo del proyecto se puedan corregir errores o fallos en el transcurso del mismo.

4.3. Herramientas de desarrollo

Se realizará la interfaz con un framework llamado Shiny, el cual está basado en el lenguaje de programación R. Shiny permite crear aplicaciones web multiplataforma interactivas y está basada en la programación reactiva que vincula datos de entrada con los de salida. Se seleccionó este framework para el desarrollo del proyecto porque esta versión de IRACE está desarrollada en R y que este lenguaje ofrece excelentes características para manipular, analizar y representar datos.

5. Gestión del Proyecto

5.1. Carta Gantt

A continuación, se presentará la planificación para el presente proyecto, la cual posee tres iteraciones y cada una contiene las fechas correspondientes a las entregas de avance, software y final. Cada iteración cuenta con definición de requerimientos, diseño, programación, ensamble, y testing.

5.1.1. Primera fase carta gantt

La primera fase cuenta con la primera reunión con el cliente y definición de problemática y solución para el proyecto.

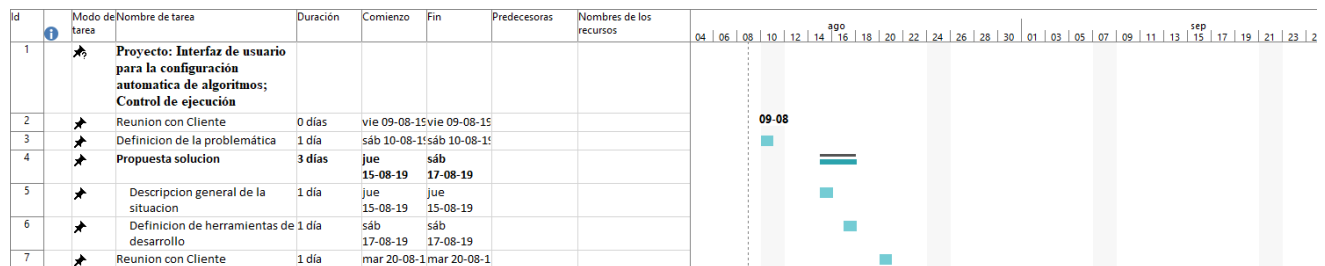


Figura 5.1 Carta Gantt primera fase

5.1.2. Primera iteración carta gantt

En la siguiente imagen, se puede visualizar las fechas de las actividades de la primera iteración junto con la fecha de la entrega de avance.

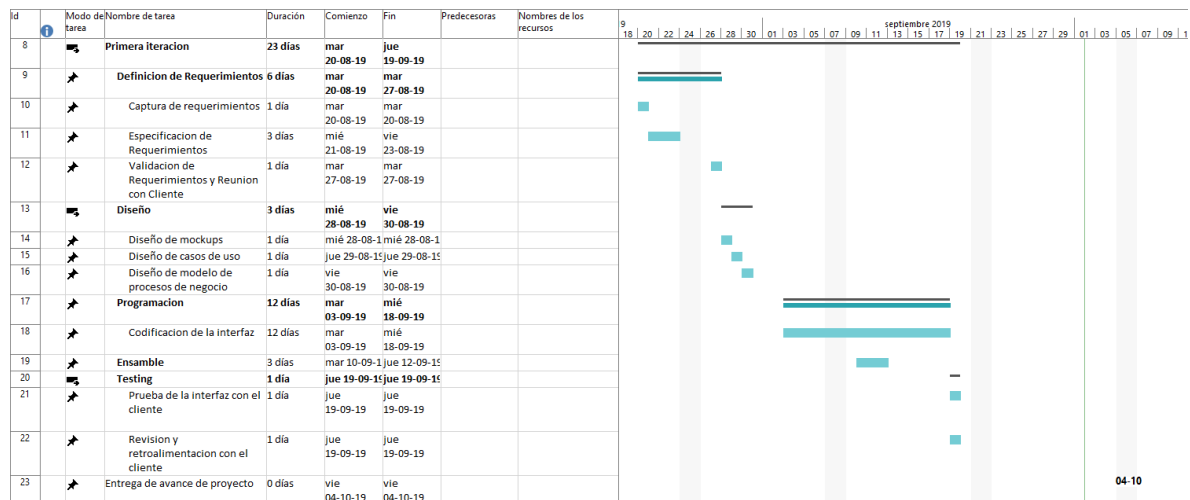


Figura 5.2 Carta Gantt primera iteración

5.1.3. Segunda iteración carta gantt

En la siguiente imagen se puede visualizar las fechas y actividades de la segunda iteración.

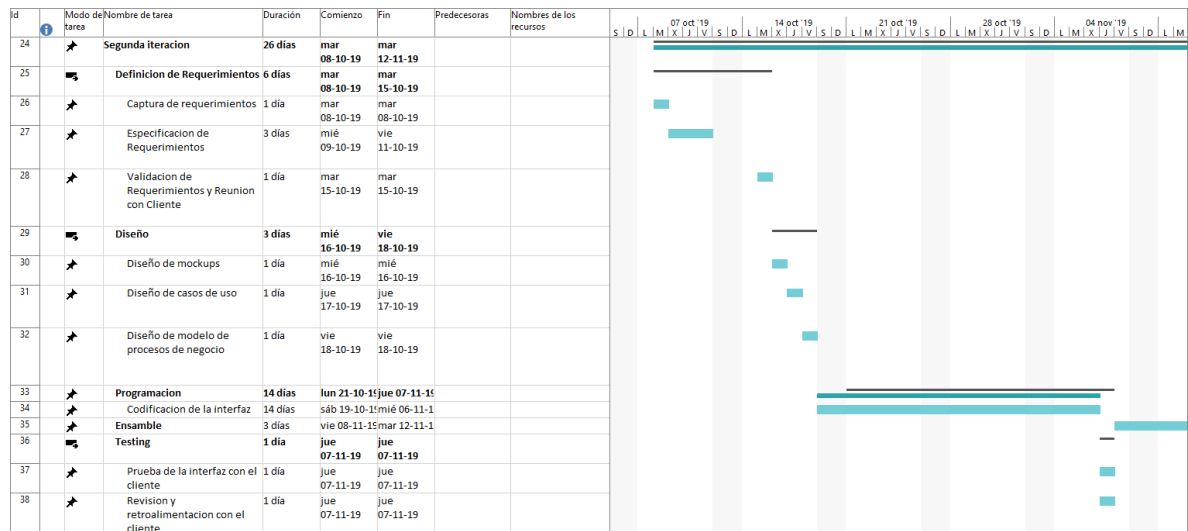


Figura 5.3 Carta Gantt segunda iteración

5.1.4. Tercera iteración carta gantt

En la siguiente imagen se puede visualizar las fechas y actividades de la tercera iteración junto con las fechas de entrega y revisión de software, entrega de informe final y presentación final.

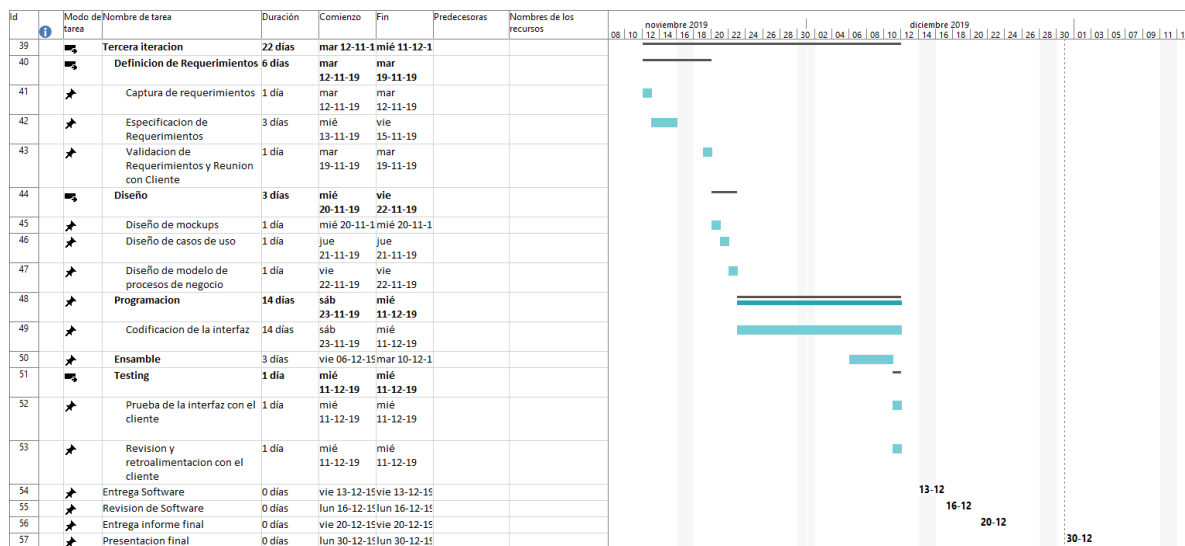


Figura 5.4 Carta Gantt tercera iteración

6. Especificación de requerimientos

6.1. Requerimientos funcionales

A continuación, se presentarán los requerimientos funcionales del proyecto. Se detallará la prioridad de estos con un 1, 2 o 3. El primero representa que el requerimiento es imprescindible para el funcionamiento del módulo de control de ejecución, el segundo que no afecta a la funcionalidad de la interfaz y el tercero que no tiene mayor relevancia y que se podría implementar a futuro.

Tabla 6.1 Requerimientos Funcionales

ID Requerimiento	Descripción	Prioridad
RF 01	El sistema deberá proveer una barra de progreso que indicará cuales son los valores que se están refrescando durante la ejecución de IRACE	1
RF 02	El sistema deberá implementar un botón que re dirccione a la sección “Summary”.	1
RF 02.1	El sistema deberá mostrar la cantidad de iteraciones que se efectuaron en la ejecución de IRACE.	1
RF 02.2	El sistema deberá mostrar la cantidad de instancias que se ingresaron en el escenario de ejecución.	1
RF 02.3	El sistema deberá mostrar el número de candidatos elite en tiempo real.	1
RF 02.4	El sistema deberá mostrar una tabla que contenga todas las configuraciones elites con los valores de sus respectivos parámetros. La tabla deberá actualizarse en tiempo real mientras se ejecuta IRACE.	1
RF 02.4.1	La tabla de configuraciones elite deberá filtrarse por iteración.	1
RF 02.5	El sistema deberá mostrar una tabla con todas las configuraciones con los parámetros y sus respectivos valores.	1
RF 02.5.1	La tabla que contenga todas las configuraciones deberá actualizarse en tiempo real mientras se ejecuta IRACE.	1
RF 03	El sistema deberá implementar un botón que re dirccione a la sección de “Performance”.	1
RF 03.1	El sistema deberá mostrar un boxplot que representará la mediana de las configuraciones elite.	1
RF 03.1.1	El boxplot deberá filtrarse según un rango de iteraciones.	1

ID Requerimiento	Descripción	Prioridad
RF 03.2	El sistema deberá mostrar un plot de convergencia que representará el número de ejecuciones del algoritmo objetivo que se está probando en la ejecución versus el valor medio sobre el conjunto de prueba.	1
RF 04	El sistema deberá implementar un botón que re direcciona a la sección “Frequency”.	1
RF 04.1	El sistema deberá implementar un plot de frecuencia de sampling que representara gráficamente los valores de los parámetros de las configuraciones elite.	1
RF 04.1.1	El plot de frecuencia de sampling deberá filtrarse por un rango de iteraciones.	1
RF 04.1.2	El usuario podrá elegir que parámetros desea representar mediante el plot frecuencia de sampling.	1
RF 04.1.3	Los plot de frecuencia de sampling deberán generarse mediante imágenes y concatenarlas para mostrarlas en la interfaz debido a que existen escenarios con una gran cantidad de parámetros, generando demasiados plots para mostrarlos en una sola pantalla.	1
RF 04.2	El sistema deberá implementar un plot llamado Parallel Coordinates que representará gráficamente mediante líneas paralelas los valores de los parámetros de las configuraciones elite.	1
RF 04.2.1	El plot Parallel Coordinates deberá filtrarse por un rango de iteraciones.	1
RF 04.2.2	Los plot Parallel Coordinates deberá generarse mediante imágenes y concatenarlas para mostrarlas en la interfaz debido a que existen escenarios con una gran cantidad de parámetros, generando demasiados plots para mostrarlos en una sola pantalla.	1
RF 05	El sistema deberá proveer un icono que re direcciona al sitio web de IRACE con la guía de usuario.	3
RF 06	El sistema deberá proveer un mensaje de estado que avise al usuario si IRACE se está ejecutando o no.	1

ID Requerimiento	Descripción	Prioridad
RF 07	El sistema deberá proveer un botón que permita al usuario detener o terminar la ejecución del algoritmo.	1
RF 07.1	Al terminar la ejecución el módulo de control deberá cerrarse y volver al menú principal.	2

6.2. Mockups

6.2.1. Control de Ejecución: Output línea de comandos

El siguiente prototipo muestra la sección de “Command Line”, la cual muestra el comando ingresado por el usuario al momento de iniciar la ejecución.

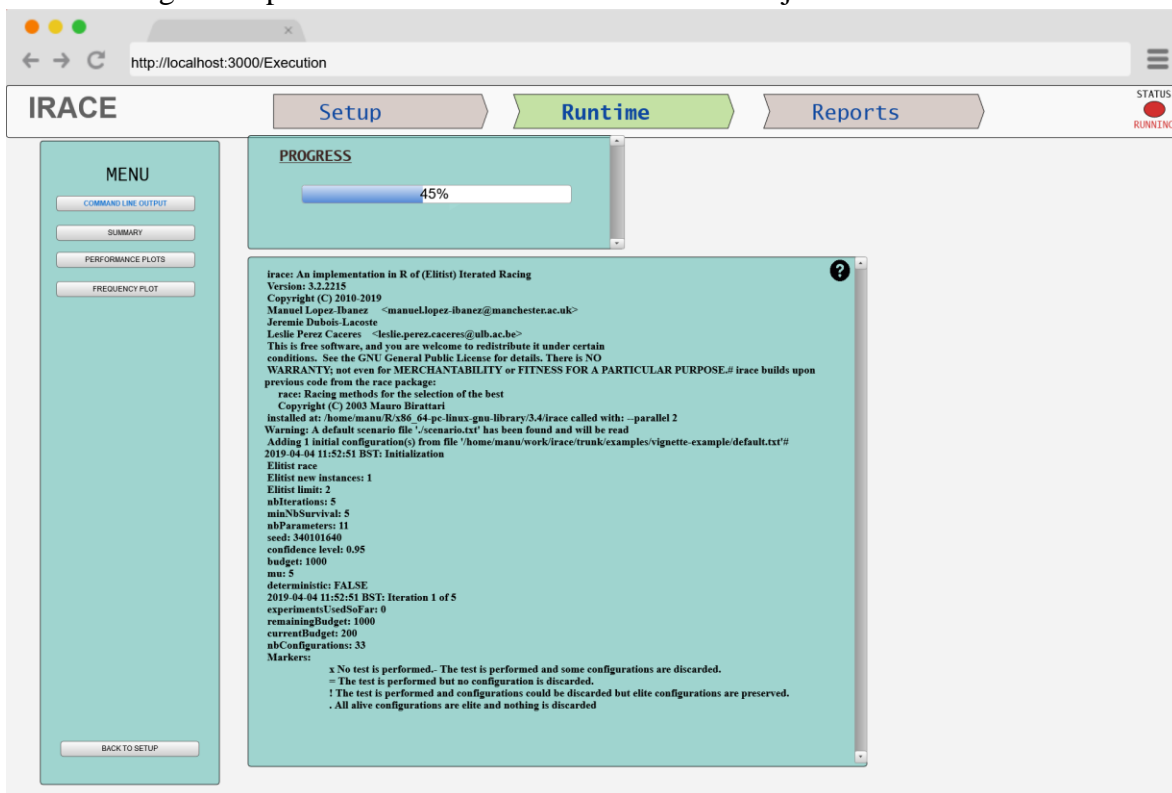


Figura 6.1 Mockup Command Line

6.2.2. Control de Ejecución: Resumen

El siguiente prototipo muestra la sección de “Summary”, en la cual se visualiza el resumen del escenario creado para la ejecución y la configuración élite, el cual va a ir cambiando progresivamente a medida que el software entregue resultados.

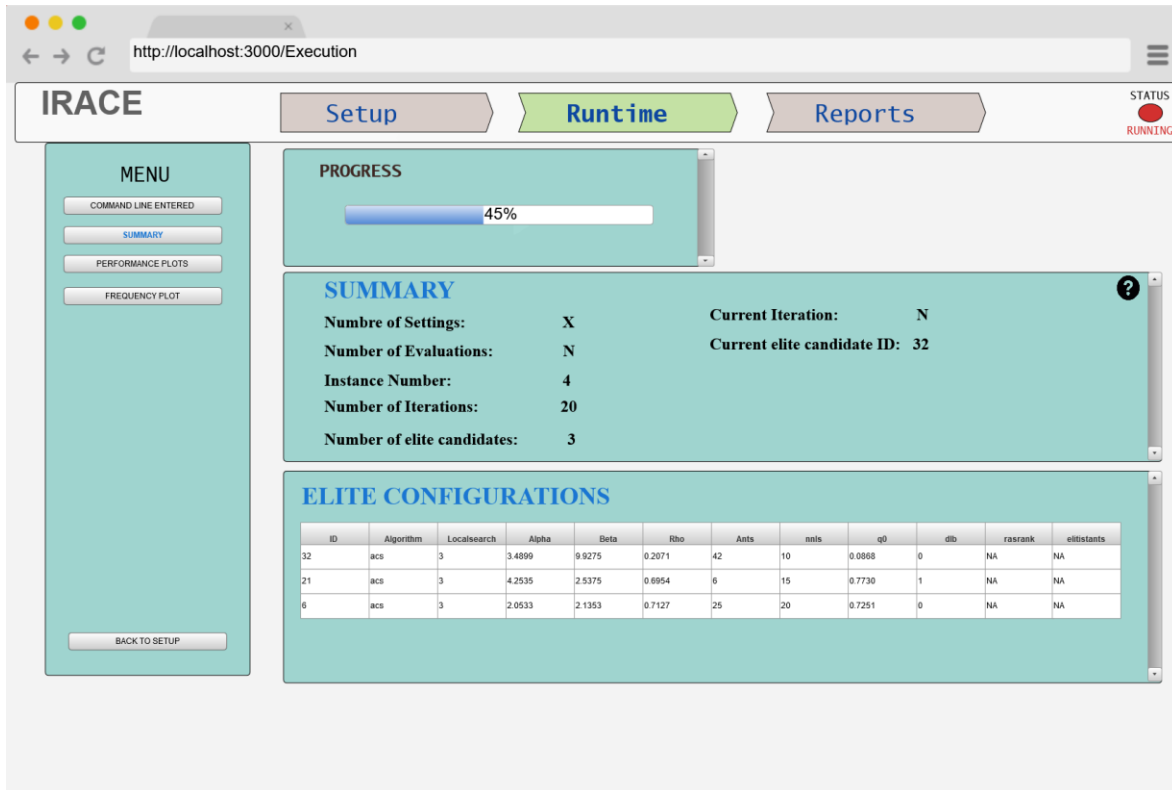


Figura 6.2 Mockup Summary

6.2.3. Control de Ejecucion: Plots de Performance

El siguiente prototipo muestra la sección “Performance Plots”, en la cual se visualiza un boxplot y un plot de convergencia. El primero representa la calidad de configuración y la configuración, mientras que el segundo representa las iteraciones y la calidad.

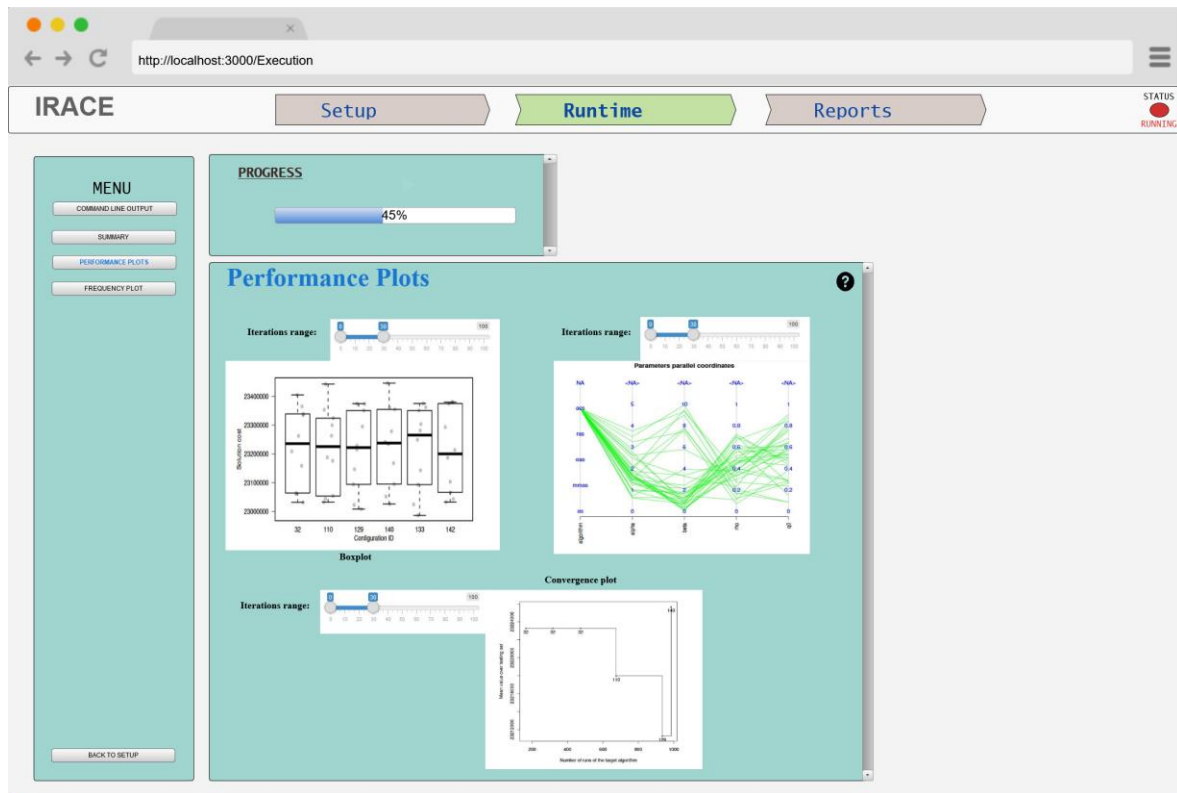


Figura 6.3 Mockup Performance Plots

6.2.4. Control de Ejecución: Plots de Frecuencia

El siguiente prototipo muestra la sección “Frequency Plots”, que permite visualizar el gráfico Parallel Coordinates y la frecuencia de sampling.

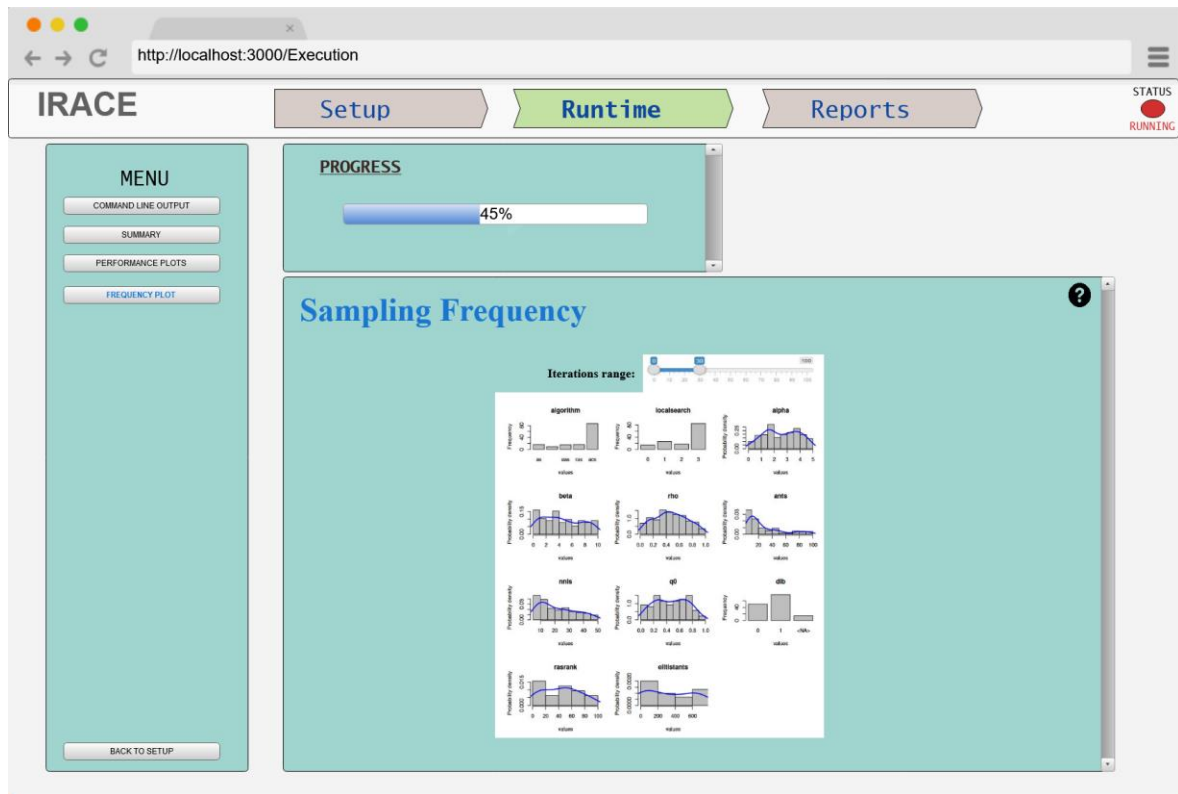


Figura 6.4 Mockup Frequency Plots

6.3. Aplicación final

6.3.1. Sección “Summary”

En la sección “Summary” se muestra un resumen de las variables que posee IRACE, las cuales son: cantidad de parámetros que poseen las configuraciones, cantidad total de configuraciones, cantidad de configuraciones elite (las cuales se pueden filtrar por número de iteración), cantidad de instancias, cantidad de instancias usadas hasta el momento, cantidad máxima de experimentos, cantidad de experimentos usados hasta el momento y la cantidad de iteraciones. La cantidad de parámetros de las configuraciones, son los parámetros que posee el algoritmo objetivo.

Por otro lado, existe una tabla donde se muestran todas las configuraciones elites con los valores de sus respectivos parámetros. Además, la tabla de configuraciones elite se pueden filtrar por el número de la iteración, mostrando las configuraciones pertenecientes a la iteración seleccionada. Esta cantidad de iteraciones están definidas en la sección “Summary”.

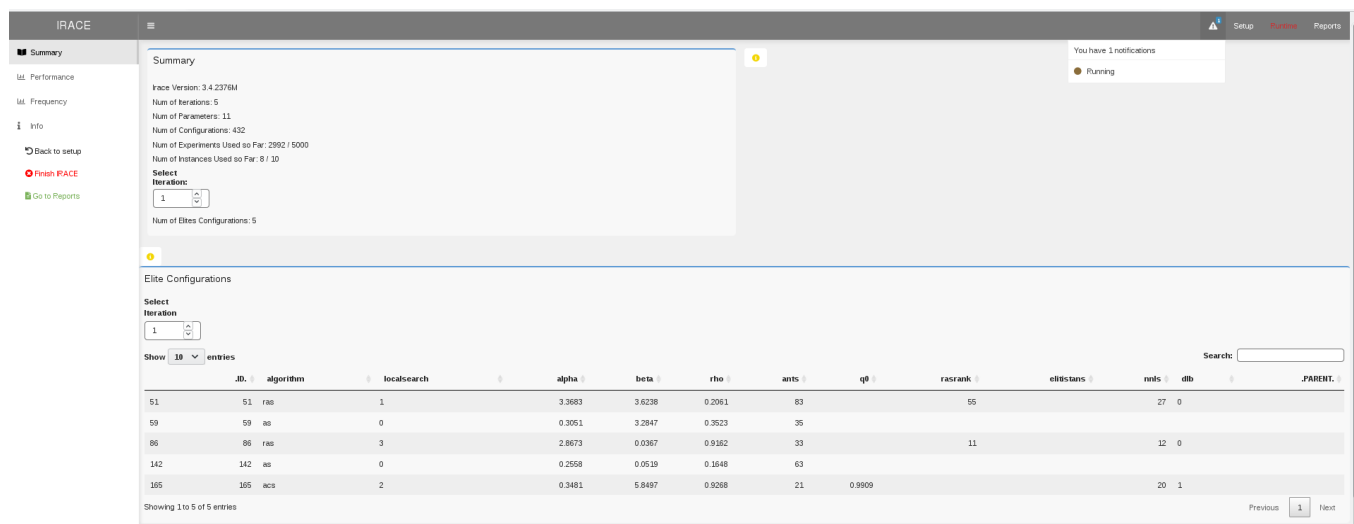


Figura 6.5 Sección Summary Aplicación Final

Por último, en la sección “Summary” se muestran en una tabla todas las configuraciones con los valores de sus respectivos parámetros.

ID	algorithm	localsearch	alpha	beta	rho	ants	q0	rasrank	elitists	nmls	dlb	PARENT
1	as	1	0.212	1.7696	0.7403	83				26	0	
2	ras	2	1.1892	3.1605	0.5524	77		77		28	0	
3	mmas	1	3.4624	3.9708	0.5886	33				31	0	
4	acs	0	4.309	9.3058	0.7372	35	0.5182					
5	eas	1	4.6441	9.5699	0.238	6			478	34	0	
6	ras	3	0.1848	1.7683	0.1688	33		60		20	0	
7	as	1	2.0332	3.3001	0.0987	73				35	1	
8	mmas	1	4.2126	7.5963	0.0729	99				47	1	
9	acs	1	1.2606	8.1675	0.9762	73	0.9542			25	0	
10	ras	0	0.8715	7.5746	0.2257	17		14				

Figura 6.6 Sección Summary 2 Aplicación Final

6.3.2. Sección “Performance”

En la sección “Performance” se muestra un BoxPlot y un plot de Performance. El primero muestra la calidad de una configuración a través de varias instancias en relación entre los parámetros de la configuración. El boxplot muestra cuartiles y la mediana de todas las configuraciones elite por un rango de iteraciones. El segundo muestra el número de ejecuciones del algoritmo objetivo o el algoritmo que se está probando en la ejecución versus el valor medio sobre el conjunto de prueba.

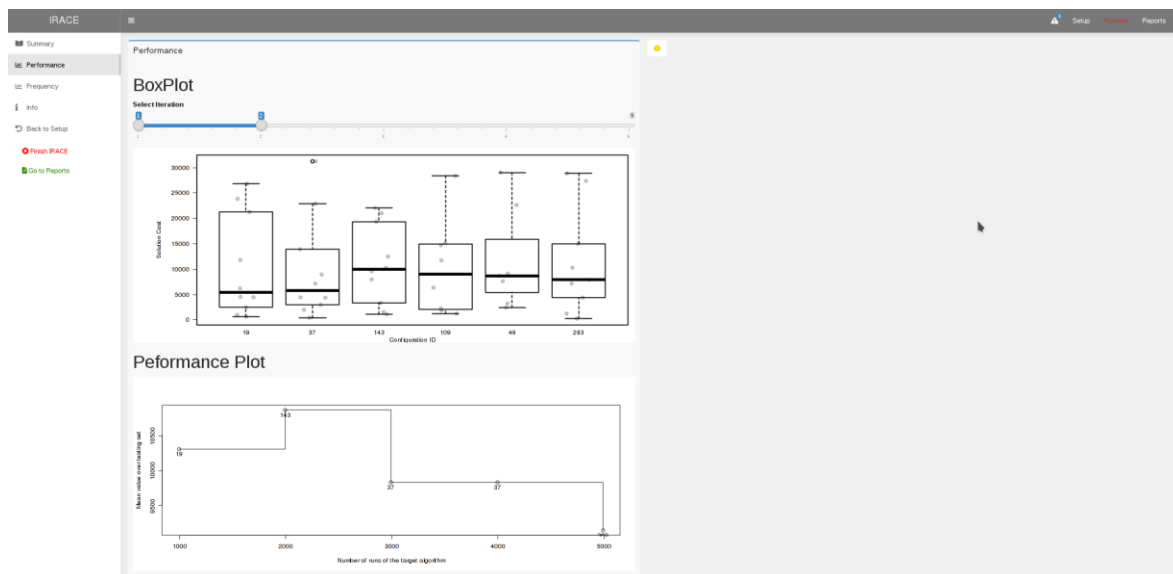


Figura 6.7 Sección Performance Aplicación Final

6.3.3. Sección “Frequency”

En la sección “Frequency” se muestran dos plots. El primero es un plot de frecuencia de sampling el cual representa gráficamente los valores de cada parámetro de las configuraciones elite de manera discreta o continua, dependiendo del tipo de parámetro. La función de este plot es ayudar al usuario a que comprenda a que valores los modelos utilizados han convergido en el proceso de optimización. Se puede definir un rango de iteraciones para filtrar los parámetros de las configuraciones elites por cada iteración.

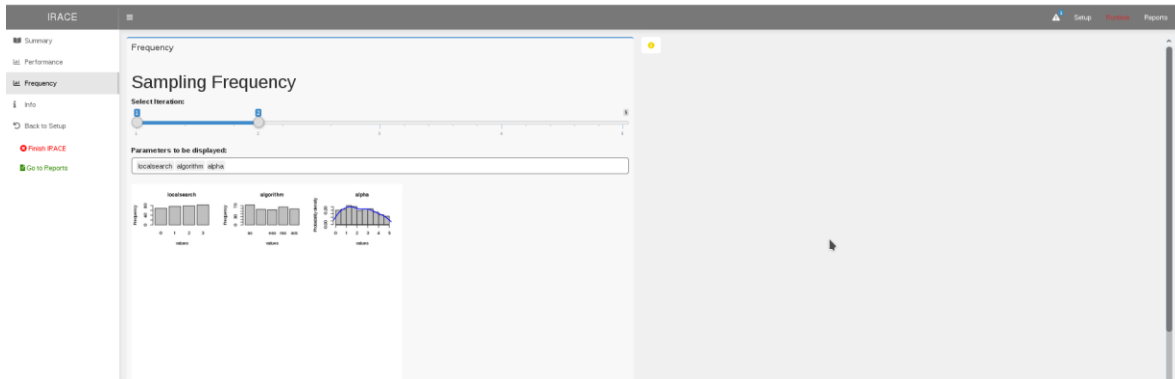


Figura 6.8 Sección Frequency Aplicación Final

El segundo es un plot llamado Parallel Coordinates representa gráficamente mediante líneas paralelas un grupo de configuraciones enfocado en la relación entre los parámetros. Esto permite que se grafiquen varias configuraciones para mostrar las tendencias generales de los valores de los parámetros de éstas. Los valores de los parámetros de las configuraciones elite son filtradas por un rango de iteraciones.

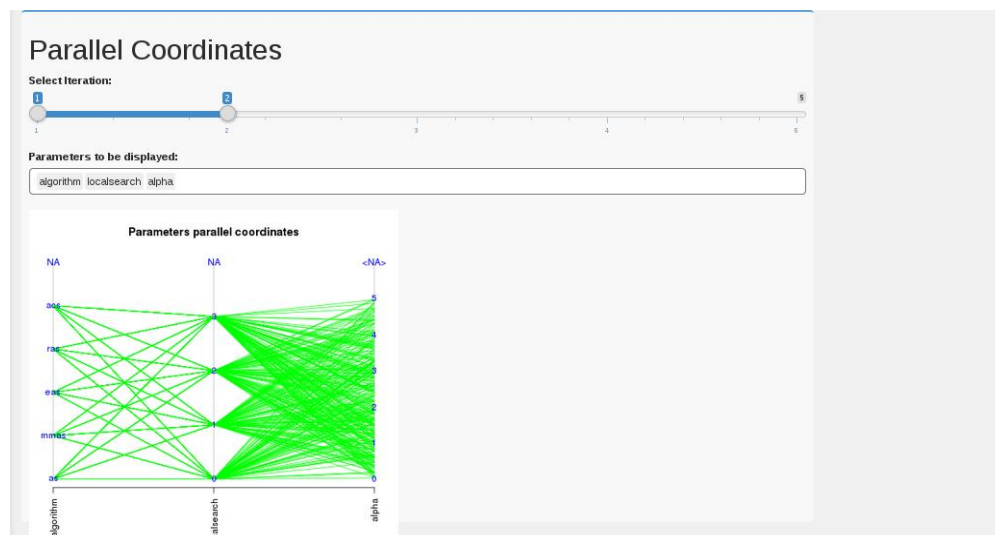


Figura 6.9 Sección Frequency 2 Aplicación Final

6.3.4. Detalles de la aplicación

La interfaz posee un sidebar en donde se puede acceder a las distintas secciones nombradas anteriormente. El botón “Info” re direccionará al usuario al sitio web del paquete de IRACE. El botón “Finish IRACE” termina la ejecución de IRACE y automáticamente cierra la aplicación de “Runtime” y re direcciona al usuario al menú principal para crear un nuevo escenario de ejecución o cargar un reporte. Por otro lado, cuando la ejecución termina, se habilita el botón “Go to Reports” para re direccionar al usuario al tercer módulo llamado “Reports”

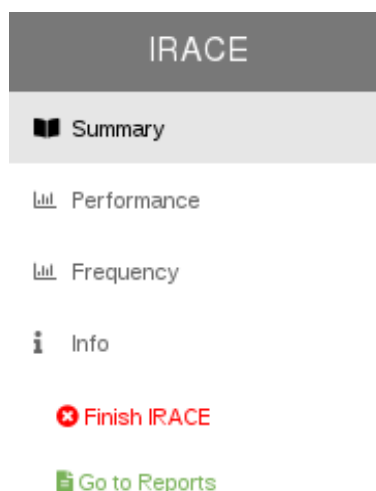


Figura 6.10 Detalles de la Aplicación SideBar

Para determinar el estado de la ejecución de IRACE, se despliega una notificación en el navbar.

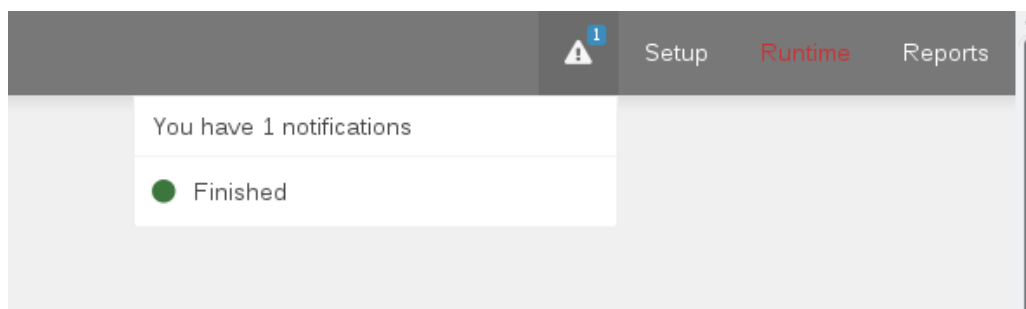


Figura 6.11 Detalles de la Aplicación Status

6.4. Casos de Uso

A continuación, se presenta un diagrama de casos de uso. Este diagrama busca explicar mediante acciones, el comportamiento del sistema o descripción de la funcionalidad de este (en este caso interfaz) desde el punto de vista del usuario independientemente de la implementación.

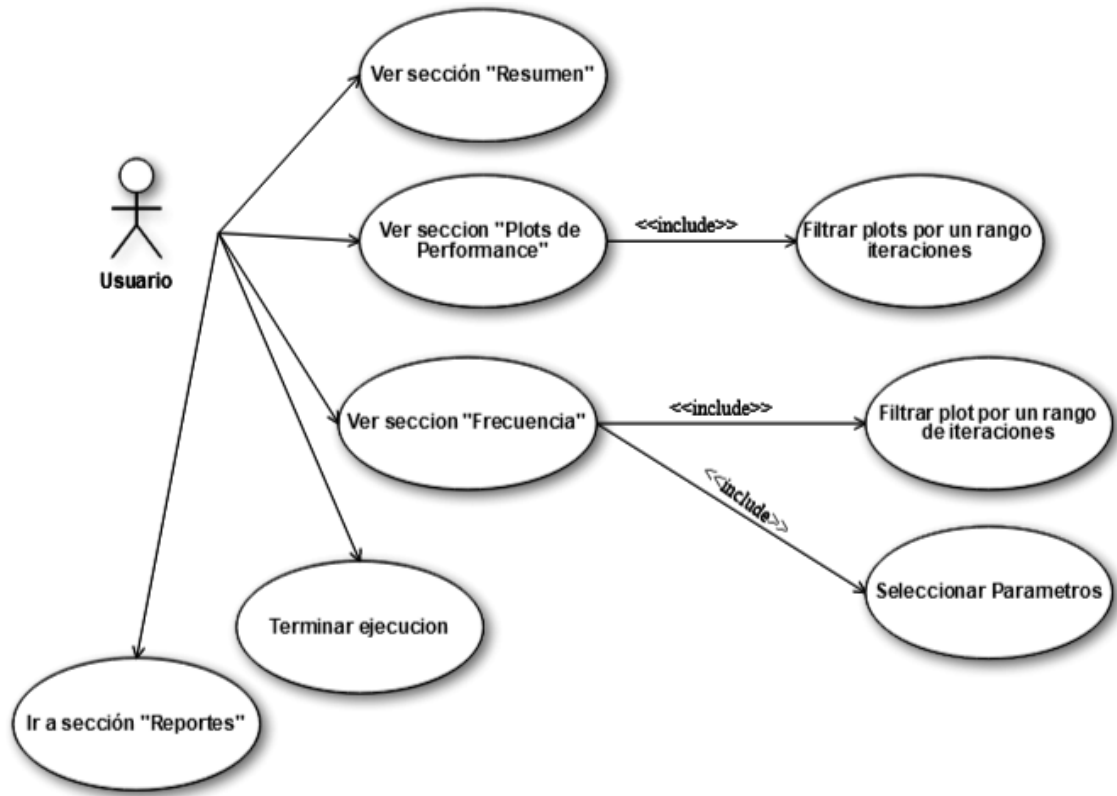
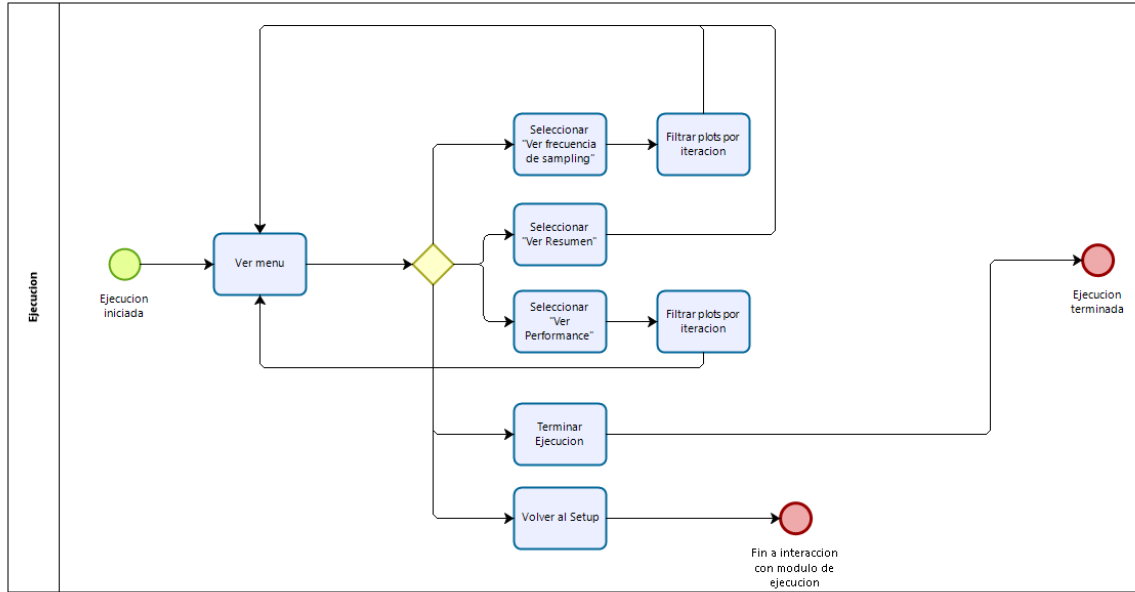


Figura 6.12 Diagrama caso de uso modulo control de ejecución

En el caso de uso “Ver sección Resumen” el usuario tendrá la posibilidad de visualizar el número de configuraciones, número de instancias, número de instancias usadas hasta el momento, número de iteraciones, número de candidatos elite y las configuraciones elite a medida que el software entregue dichos datos. Por otro lado, en el caso de uso “Ver sección Plots de performance” el usuario podrá visualizar un boxplot que representa a las configuraciones en el eje x y la calidad de configuración en el eje y. Por último, está un plot de convergencia que muestra las iteraciones en el eje x y la calidad en el eje y. Finalmente, en el caso de uso “Ver sección de Frecuencia” se mostrara un plot que representa a parámetros categóricos o numéricos en el eje x y la frecuencia en el eje y. En el caso de que los parámetros sean categóricos, el grafico será un gráfico de barra, en el caso contrario será un gráfico de área. Por último, en la sección de frecuencia, el usuario podrá visualizar el plot Paralel Cordinates que representa a los mejores candidatos y muestra las configuraciones en el eje x y el valor de los parámetros en el eje y

6.5. Modelo de procesos de negocio

A continuación, se presentará el modelo de procesos de negocio del proyecto. En este se representa el flujo de tareas que el usuario puede realizar con la interfaz de control de ejecución.



Powered by
bizagi
Modeler

Figura 6.13 Modelo de procesos de negocio modulo control de ejecución

En el siguiente modelo se puede apreciar el flujo de tareas que puede realizar el usuario dentro de la interfaz del módulo control de ejecución. Cuando la ejecución comienza, el usuario tendrá un menú a su disposición en el cual podrá revisar el resumen de la ejecución, plots de frecuencia de sampling y plots de performance. Cuando el usuario termina la ejecución, el usuario es re direccionado al menú principal.

7. Estudio de factibilidad

7.1. Factibilidad Técnica

El estudio de la factibilidad técnica es una evaluación de la facultad del sistema para determinar si será funcional y podrá mantenerse en el tiempo. En este caso se realizará un estudio de factibilidad técnica enfocado al área de informática, describiendo los recursos que se requerirán tales como recurso humano, hardware y software, para el desarrollo y funcionamiento del sistema.

Tabla 7.1 Tabla de factibilidad técnica

Tipo de recurso	Nombre del recurso	Descripción
Recurso Humano	Ingeniero Informático	En este proyecto se necesitará de un ingeniero informático que cumpla la función de analista y de desarrollador. Éste deberá tener la capacidad de abarcar desde la captura de requerimientos y modelado del sistema hasta la programación de la interfaz interactiva para IRACE.
Hardware	Computadores	Para el desarrollo del proyecto se precisará de computadores que puedan realizar las tareas de programación y diseño del sistema.
Software	<ol style="list-style-type: none"> 1. Shiny 2. RStudio 3. Linux 4. Windows 10 Pro 5. Microsoft Office 365 6. Bizagi Modeler 7. Cacao 8. Google Drive Personal 9. Github Pro 	<ol style="list-style-type: none"> 1. Se utilizará el framework Shiny para la implementación de la interfaz gráfica para IRACE 2. Se utilizara RStudio para la codificación. 3. Durante el proceso de programación, se utilizara el sistema operativo Linux para este proceso del proyecto. 4. Para la realización de la documentación, modelado, presentaciones e informes se utilizará el sistema operativo Windows 10. 5. Se utilizará Microsoft Office 365 para la realización de informes, presentaciones y gestión del proyecto.

Tipo de recurso	Nombre del recurso	Descripción
		6. Para la realización del modelo de procesos de negocio se utilizara Bizagi Modeler 7. Para la realización del diagrama de casos de uso se utilizara la herramienta web Cacao. 8. Para el respaldo de documentos se utilizará la herramienta web Google Drive. 9. Para el guardado y control del código en el proceso de programación se utilizara la herramienta web Github

7.2. Factibilidad Económica

En el estudio de la factibilidad económica se determina el presupuesto de costos de los recursos técnicos detallados anteriormente. Al realizar este análisis se podrá determinar si desarrollar el proyecto es factible económicamente.

7.2.1. Recurso Humano

Tabla 7.2 Factibilidad económica recurso humano

Tipo	Cantidad	Valor en CLP por Hora Aproximado	Duración del proyecto	Total Proyecto (CLP) Aproximado
Analista	1	\$4.180	720 Hrs	\$3.000.0000
Programador	1	\$5.231	360 Hrs	\$1.890.000
				Total: \$4.890.000

7.2.2. Hardware

Tabla 7.3 Factibilidad económica hardware

Tipo	Cantidad	Valor Individual Aproximado (CLP)	Total Proyecto (Aproximado) en CLP
Computadores para el desarrollo	1	\$300.000	\$300.000
			Total: \$300.000

7.2.3. Software

Tabla 7.4 Factibilidad económica software

Tipo	Cantidad	Valor CLP(Aproximado)	Total en CLP (Aproximado)
Shiny	1	Gratuito	Gratuito
RStudio	1	Gratuito	Gratuito
Linux	1	Gratuito	Gratuito
Licencia Windows 10 Pro	1	\$179.999	\$179.999
Licencia Microsoft Office 365	1	\$5.699/Mes	\$22.796
Bizagi Modeler Profesional	1	\$5800/Mes	\$11.600
Licencia Cacao	1	\$3600/Mes	\$7.200
Google Drive Personal	1	Gratuito	Gratuito

Tipo	Cantidad	Valor CLP(Aproximado)	Total en CLP (Aproximado)
Github Pro	1	\$5000/Mes	\$10.000
			Total:\$231.595

7.3. Factibilidad Operacional

Actualmente, el software IRACE no presenta una interfaz interactiva en donde el usuario pueda visualizar de manera amigable y ordenada los resultados que este le entrega por lo que la factibilidad de que la interfaz de usuario sea utilizada es muy alta.

En esta primera fase de recolección de requerimientos, estos fueron especificados y validados. Esto quiere decir que a medida que se den las reuniones con el cliente durante el proceso de desarrollo, los requisitos son susceptibles de cambios.

Según el análisis previo, es factible el desarrollo de la interfaz IRACE, ya que, por un lado, es requerido por una gran cantidad de usuarios que ocupan el software y que las herramientas para la implementación son gratuitas, lo que implica bajos costos.

7.4. Factibilidad Legal

La factibilidad legal nos permite determinar los derechos que tienen los autores sobre la documentación realizada por estos en este proyecto. También señala cuales son las leyes, según la Legislación Informática de la República de Chile, que se deben cumplir para todo proyecto informático. El software IRACE esta patentado en el registro público de propiedad de derechos de autor. Para el presente proyecto, el cliente, desarrollador del software, dará acceso al desarrollador para la implementación de la interfaz.

Se identificarán las que se deben cumplir para este proyecto.

- Ley n° 19.223. Tipifica figuras penales relativas a la informática.
- Ley n° 17.336, Sobre propiedad intelectual.

8. Desarrollo de la aplicación

La interfaz desarrollada con el framework Shiny, tiene la siguiente estructura: El template o diseño de la interfaz, esta creado en base a HTML, mientras que en el lado del servidor se utiliza el lenguaje R para las funciones, outputs e inputs de datos.

Actualmente, los tres distintos módulos: Setup, Runtime y Reports están integrados para que funcionen linealmente desde el setup hasta el último módulo de reportes. El módulo de control es ejecutado inmediatamente cuando el usuario inicia la ejecución de IRACE en el módulo del Setup.

Como primera fase del desarrollo de la aplicación, se tuvo que estudiar de manera breve y continuada el lenguaje de programación R, el cual es el que se usó junto con el framework Shiny para el desarrollo de la interfaz. Por otro lado, se tuvo que estudiar la manera en que se comportaba el framework con dicho lenguaje y buscar las herramientas que este proveía para desarrollar la solución deseada. En el módulo de control lo importante es leer el archivo “.Rdata” que IRACE entrega durante su ejecución, ya que es el que contiene todos los datos esenciales para el usuario, por lo tanto, se optó por usar los outputs que facilita R para mostrar los datos que conformarían el “Summary”, la tabla de “Elites Configurations” y la tabla “All Configurations” en la sección de resumen, los plots de Convergencia y Boxplot, en la sección de “Performance” y Parallel Coordinates y plot de Frecuencia en la sección de “Frequency”.

Como segunda fase, se debía buscar un método para que la información mostrada en la interfaz se refrescara en tiempo real mientras que la ejecución de IRACE se realizaba en segundo plano y que ésta sea representada por una barra de progreso, por lo que se tuvo que utilizar los eventos reactivos de Shiny juntos con una función que permitía que los outputs que mostraban la información, cambiaran a medida que el archivo “. Rdata” se actualizaba.

Para realizar la codificación se utilizó Rstudio, ya que esta plataforma permite consultar la documentación de R en la misma herramienta, además de instalar y mostrar la documentación de los paquetes que proporciona R.

9. Expectativas a futuro

Actualmente, las tres interfaces desarrolladas están integradas para que funcionen de manera lineal partiendo desde el Setup, pasando por el módulo de Runtime y terminando en el módulo de reportes, pero aún son aplicaciones separadas. En base a lo anterior, se esperaría que el desarrollo de la interfaz continuara luego de finalizado este proyecto, unificando los tres módulos para lograr mayor fluidez y optimizar líneas de código.

Finalizado el desarrollo de la interfaz para IRACE, se podría decir que los requerimientos se cumplieron y que el módulo de control cumple con su función de mantener al usuario informado en tiempo real de los datos que IRACE entrega durante su ejecución. Por otro lado, aun quedaron detalles en la interfaz que no afectan el propósito real del módulo pero que son importantes resaltar para que en un futuro se mejoren o se supriman.

Partiendo por lo más general, se debería revisar la forma en la que esta ordenada la información que se obtiene del archivo de IRACE para mostrar los datos de manera más coherente, con una distribución más lógica de la información.

Por otro lado, se debería volver a implementar la barra de progreso para que ésta esté basada en una variable propia de IRACE y que sea continua, ya que la que está implementada actualmente es una “simulación” de avance a medida que los outputs muestran la información.

Como se dijo anteriormente, aún existen detalles que pueden disminuir la calidad de la aplicación, sin afectar su funcionamiento. Por ejemplo, se debería añadir un botón que vuelva al “Setup” y otro que permita al usuario volver al menú principal, ya que actualmente ésta acción solo puede realizarse cuando el usuario finaliza por su cuenta la ejecución de IRACE.

Por último, la interfaz está desarrollada para que funcione en el sistema operativo Linux, ya que posee funciones que leen archivos, ingresan comandos a la terminal, que son tareas que Windows podría invalidar, por lo que para futuros arreglos se debería aprovechar la particularidad de R, que es multiplataforma, para que sea compatible también en Windows.

10. Conclusión

Actualmente, el software IRACE es utilizado por usuarios de todo el mundo por lo que la importancia de que éste posea una interfaz gráfica que permita una interacción amigable con el software, ya sea en el ingreso de parámetros y visualización de resultados, es enorme.

El desarrollo del proyecto fue un desafío ya que se tuvo que aprender un lenguaje de programación totalmente desconocido, pero se consiguió crear una interfaz funcional que cumple con todos los requerimientos fijados por el cliente durante todo el proceso de desarrollo.

Se optó por un modelo iterativo para mantener un contacto semanal con el cliente para revisiones, arreglar fallos en el desarrollo y aclarar dudas en el transcurso de la realización del proyecto. Se puede decir que este modelo contribuyó en gran parte a que el desarrollo de la interfaz y del proyecto en general lograra llegar hasta esta fase final.

Como fue explicado anteriormente, IRACE es un software de optimización. Es muy común que estos no posean una interfaz interactiva, ya que son complejos de implementar y por lo general son utilizados para resolver un problema específico. Según el estudio que se realizó hasta el final de este proyecto, la interfaz contemplará todos los resultados entregados por el software de manera amigable para el usuario y en tiempo real.

Según el modelo de proceso de desarrollo definido, se cumplió el método de reuniones semanales para informar, reportar avance y/o dudas acerca del funcionamiento de IRACE. Lo que aportó dinamismo al desarrollo y optimizar la depuración de la aplicación.

Para finalizar el presente informe, se destacará que existió un acuerdo por parte del cliente y del desarrollador que lo especificado en los puntos anteriores está correcto. Se espera que, al finalizar el proyecto, éste pueda ser retomado para solucionar pequeños detalles y/o mejorar o agregar funcionalidades nuevas.

Referencias

- Lopez-Ibáñez, M., Dubois-Lacoste, J., Pérez, L., Stützle, T., & Birattari, M. (s.f.) et al., The irace Package: Iterated Race for Automatic Algorithm Configuration. Recuperado 2 octubre, 2019, de
- López, M., Dubois, J., Pérez, L., Birattari, M., Stützle, T.. (2016). The irce package: Iterated racing for automatic algorithm configuration. septiembre 9, 2019, de sciencedirect Sitio web: <https://www.sciencedirect.com/science/article/pii/S2214716015300270?via%3Dihub>
- López, M.. (2011). The irace Software Package. septiembre 9, 2019, de iridia Sitio web: <http://iridia.ulb.ac.be/irace/files/irace-comex-tutorial.pdf>
- Neuvo. (s.f.). Salario en Chile. Recuperado de <https://neuvo.cl/remuneracion/>
- Microsoft. (s.f.). Microsoft Store. Recuperado de <https://www.microsoft.com/es-cl/store/b/windows>
- Microsoft. (s.f.-b). Office. Recuperado de <https://products.office.com/es-cl/buy/office>
- Bizagi. (s.f.). Bizagi Modeler Software de modelamiento de procesos de negocios (BPM).
- Recuperado de <https://www.bizagi.com/es/productos/bpm-suite/modeler>
- Github. (s.f.). Github Plans for every developer. Recuperado de <https://github.com/pricing>
- Nulab. (s.f.). Cacao. Recuperado de <https://cacao.com/es/pricing>