



Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1-CC
Proyecto “Valoración de Opciones” by SoftWarrior Team
2018-08-26

Integrantes:

Nombres y Apellidos	Email	ROL USM
Paul Rojas Ojeda	paul.rojas@sansano.usm.cl	201573063-6
Beatriz Segura Pastén	beatriz.segura@sansano.usm.cl	201573041-5
Gabriela Sepúlveda Bravo	gabriela.sepulvedab@sansano.usm.cl	201573012-1

Contenido del Informe a Entregar

1. Requisitos clave (Final)	3
2. Árbol de Utilidad (Final)	4
3. Modelo de Software (Final)	5
4. Trade-offs entre tecnologías (final)	6
5. Deuda técnica incurrida	8

1. Requisitos clave (Final)

A continuación se presentan las tablas correspondientes a los Requisitos Funcionales y Extra-Funcionales finales.

Tabla 1: Requisitos funcionales (actualizados)

Req. funcional	Descripción y medición (máximo 2 líneas)
El sistema permitirá al usuario ingresar un archivo con los datos a analizar	El usuario puede ingresar un archivo con extensión CSV con los datos a trabajar con un tamaño máximo 1MB y formato adecuado.
El usuario ingresará la empresa, el intervalo de maduración, precio de ejecución, número de iteraciones y la tasa de riesgo.	El usuario debe ingresar los parámetros solicitados con un máximo de tiempo de maduración 10 años.
El sistema entregará el resultado de la valoración de opciones europeas sobre las acciones solicitadas.	Se entregará un resultado numérico para la valoración de opciones europeas sobre la acción solicitada en base a los teoremas de Black-Scholes y la simulación Monte Carlo.
El programa permitirá la visualización de gráficos.	La fluctuación de la valoración de opciones se podrá visualizar en un gráfico 2D.
Se extraerán los datos de "Yahoo! Finance"	Todos los datos solicitados por el usuario se extraerán del sitio web de Yahoo! Finance para tener fidelidad de los datos.

Tabla 2: Requisitos extra-funcionales (actualizados)

Req. extra-funcional	Descripción y medición (máximo 2 líneas)
El programa debe ser un .exe capaz de ejecutarse en Linux y en Windows	Se programará en el lenguaje Python, el cual mediante una librería específica (Py2Exe) nos permitirá realizar un ejecutable del programa.
El sistema utilizará librerías de python	Para calcular la valoración de opciones sobre los datos se hará uso de las librerías proporcionadas en python como pandas, matplotlib y numpy.
El sistema cifrará los datos extraídos de "Yahoo! Finance" y/o cargados por el usuario	Todos los datos extraídos del sitio web Yahoo! Finance y/o cargados por el usuario, serán cifrados para evitar que estos sean manipulados por terceros.

2. Árbol de Utilidad (Final)

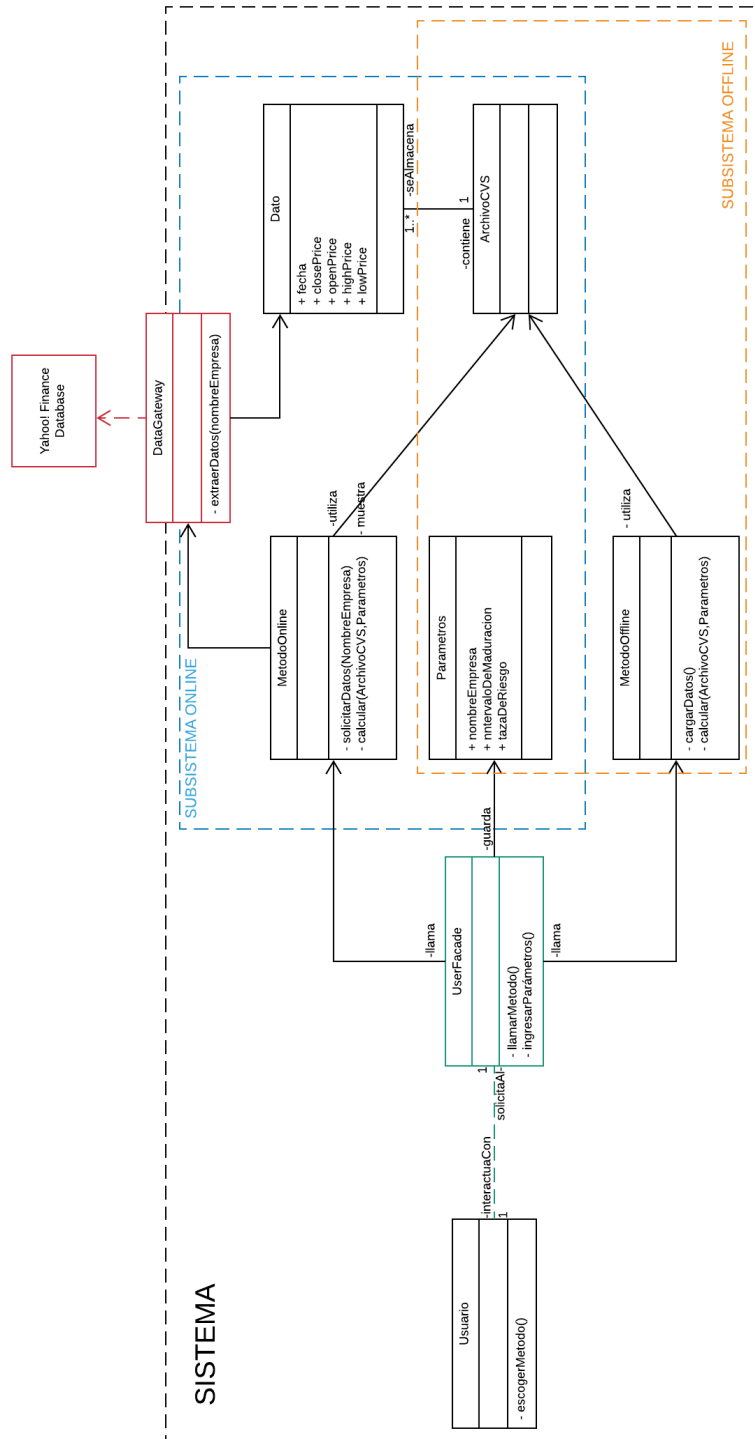
A partir de la segunda entrega, el árbol de utilidad no sufrió ningún cambio por ende a continuación se presenta el árbol de utilidad final:



Ilustración 1:Árbol de Utilidad actualizado

4. Modelo de Software (Final)

Se presenta el modelo de software final actualizado, se quitaron cosas que no pudieron ser implementadas por falta de tiempo.



5. Trade-offs entre tecnologías (final)

En la “Figura 4” se podrá encontrar el SIG (Softgoal Interdependency Graphs) final de los trade-offs que se determinaron luego de analizar los requisitos extra-funcionales y sus escenarios.

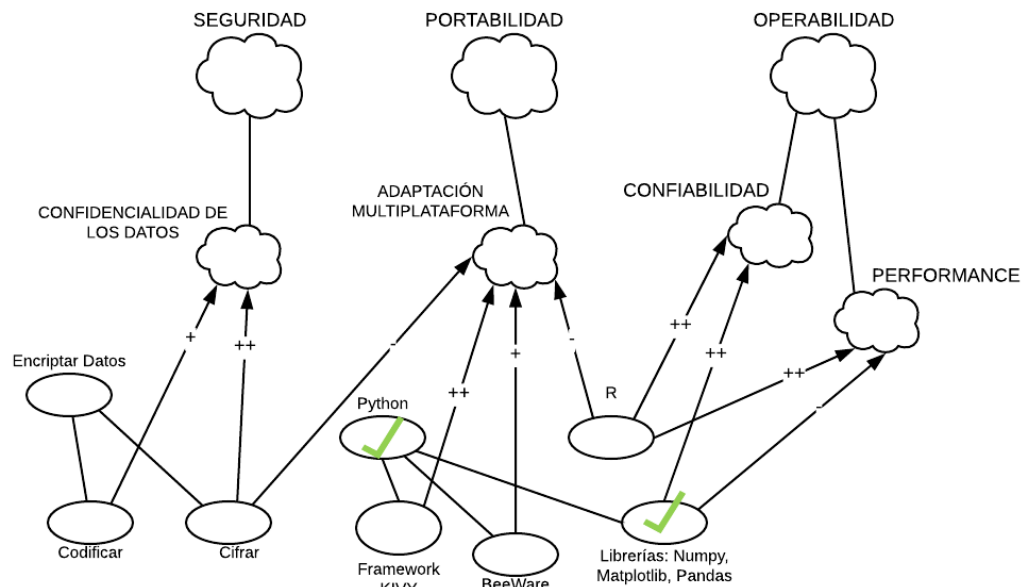


Ilustración 4: SIG de los Trade-offs

En la tabla a continuación (Tabla 4) podrá encontrar las decisiones tomadas junto con su razonamiento.

Tabla 4: Trade-offs entre opciones tecnológicas

Decisión	Softgoal	Evaluación	Razonamiento
Cifrar Datos	Confidencialidad	++	Al aplicar un algoritmo de cifrado en los datos a analizar, evitamos que terceras personas puedan intervenir en éstos asegurando la confidencialidad de los datos.
Cifrar Datos	Adaptación Multiplataforma	-	Si el algoritmo de cifrado de los datos no se implementa de forma correcta y genérica, podría causar problemas al exportar el proyecto a otras plataformas. Entiéndase que actualmente se está testeando en Linux y Windows.
Codificar Datos	Confidencialidad	+	Al realizar sólo una conversión de los datos a analizar, si bien aumentamos

			en un grado la confidencialidad de los datos, se corre el riesgo de que la tercera persona pueda adivinar de forma fácil el tipo de codificación y así pueda vulnerar los datos
Utilizar Python+Framework Kivy	Adaptación	++	Kivy nos proporciona un framework para trabajar en Python y de esta manera poder desarrollar aplicaciones que se adapten a distintas plataformas. Además de ser un framework open-source, la comunidad que lo utiliza es bastante amplia teniendo una documentación amplia para consultar.
Utilizar Python+Herramientas de BeeWare	Adaptación	+	BeeWare es una colección de herramientas que nos permite desarrollar en Python aplicaciones para distintas plataformas. También es un proyecto open-source, pero la comunidad no es tan amplia como en Kivy lo que resulta en no encontrar tanta documentación. Otros de sus contras es que al ser herramientas aisladas unas de otras, se debe aprender a trabajar en cada una de las que se necesite.
Python + librerías (Pandas, Numpy, etc)	Confiabilidad	++	La librerías que ofrece Python para el tratamiento de datos (Específicamente en matemáticas financiera), como lo son Pandas y Numpy, obtiene resultados que son aceptados de buena manera por la comunidad, lo que ha hecho que en los últimos años este lenguaje vaya ganando terreno por sobre R.
Python + librerías (Pandas, Numpy, etc)	Performance	-	Si bien la performance que desempeña este lenguaje es aceptable, aún corre con un poco de desventaja frente a R si ésta no implementa de manera correcta.
Lenguaje R	Confiabilidad	++	Reconocida es la confiabilidad de los resultados que se pueden obtener con este lenguaje, los cuales son alabados por la comunidad que se dedica a la matemática financiera.

Lenguaje R	Performance	++	También se reconoce dentro de la comunidad el performance que ofrece R. De hecho éste es uno de sus atributos más alabado.
Lenguaje R	Adaptación	-	Pero R no sólo es un lenguaje que implica tomarse un tiempo para entenderlo, si no que su adaptabilidad con otros lenguajes no es tan potente. Las librerías que permiten comunicarnos con otros lenguajes para desarrollar aplicaciones multiplataforma no siempre están bien documentadas.

6. Deuda técnica incurrida

A continuación, se presentan las deudas técnicas de nuestro proyecto, muchos de los requisitos no pudieron ser implementados por la principal causa que es la falta de tiempo.

Tabla 5: Deuda técnica

Ítem deuda técnica	Razonamiento	Impacto
Modularización del Código	Falta de tiempo.	Requiere mucho tiempo refactorizar.
Utilización de R	Complejidad en la ejecución multiplataforma de este lenguaje (problemas de compatibilidad en Windows)	Los datos no son tan precisos.
Implementación de Kivy	Falta de tiempo	El software podría no ser multiplataforma y es poco escalable.
Testing de compatibilidad	Falta de tiempo	El sistema puede no funcionar en diferentes tipos de entornos.
Testing basado en las salidas	Al utilizar números aleatorios los resultados arrojados reiteradas veces no serán los mismos.	Los resultados debían ser acotados para demostrar confiabilidad, por lo que no aseguramos que el resultado sea correcto.
Gráfico	Falta de tiempo	Resta funcionalidad al software debido a ser un requisito con prioridad alta para el cliente

Encriptación de datos	Falta de tiempo	Resta seguridad al sistema
-----------------------	-----------------	----------------------------