



Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1-CC
Proyecto “Valoración de Opciones sobre Acciones”
26/08/2018

Integrantes:

Nombres y Apellidos	Email	ROL USM
José Luis Gallardo P.	jose.gallardo.14@sansano.usm.cl	201430003-4
Marcelo González H.	marcelo.gonzalez.14@sansano.usm.cl	201430028-K
Esteban Jara C.	esteban.jara.13@sansano.usm.cl	201330002-2

Contenido del Informe a Entregar

<i>Contenido del Informe a Entregar</i>	<i>2</i>
1. Requisitos clave (Final)	3
2. Árbol de Utilidad (Final)	3
3. Modelo de Software (Final)	3
4. Trade-offs entre tecnologías (final)	3
5. Deuda técnica incurrida	3

1. Requisitos clave (Final)

Como no hubo visita por parte del cliente en las últimas dos presentaciones del proyecto, y considerando lo claro que quedaron en la presentación y posterior reunión con el mismo, los requisitos tanto funcionales como no funcionales son los mismos que los presentados para el entregable anterior.

Estos se presentan en las tablas 1 y 2, respectivamente.

Req. funcional	Descripción y medición
Generación de trayectorias	El sistema debe permitir generar simulaciones que obtienen posibles valores para las opciones de acción en forma de trayectorias.
Selección de parámetros para análisis de acciones	El sistema debe permitir la selección de la acción, periodo de tiempo, tipo, trayectorias y número de trayectorias para obtener posibles valores de las acciones.
Recopilación de valores de acciones	El sistema debe guardar un historial con el valor de las acciones consultadas.

Tabla 1: Requisitos Funcionales (FR).

Req. extra-funcional	Descripción y medición (máximo 2 líneas)
Serialización	Serialización para separar información de cada usuario
Importación de datos (online como offline)	El sistema debe permitir la importación de datos financieros desde un archivo .csv o la API de Yahoo Finance.
Gráficos de comportamiento de acciones	El sistema debe mostrar un gráfico con el valor de la acción en función del tiempo.
Gráficos de valoración y trayectorias	El sistema debe mostrar un gráfico de valoración de acción v/s número de trayectorias utilizadas.
Plataforma “Desktop”	El sistema debe ejecutarse en un ambiente “Desktop” y sin la necesidad de conectarse a internet para funcionar.
Tiempo de respuesta	Tiempo de respuesta de consultas < 20s
Desempeño	El sistema debe realizar los cálculos numéricos utilizando el lenguaje de programación R

Tabla 2: Requisitos No Funcionales (NFR).

2. Árbol de Utilidad (Final)

Considerando las correcciones del entregable anterior, el árbol de utilidad final se presenta en la figura 1.

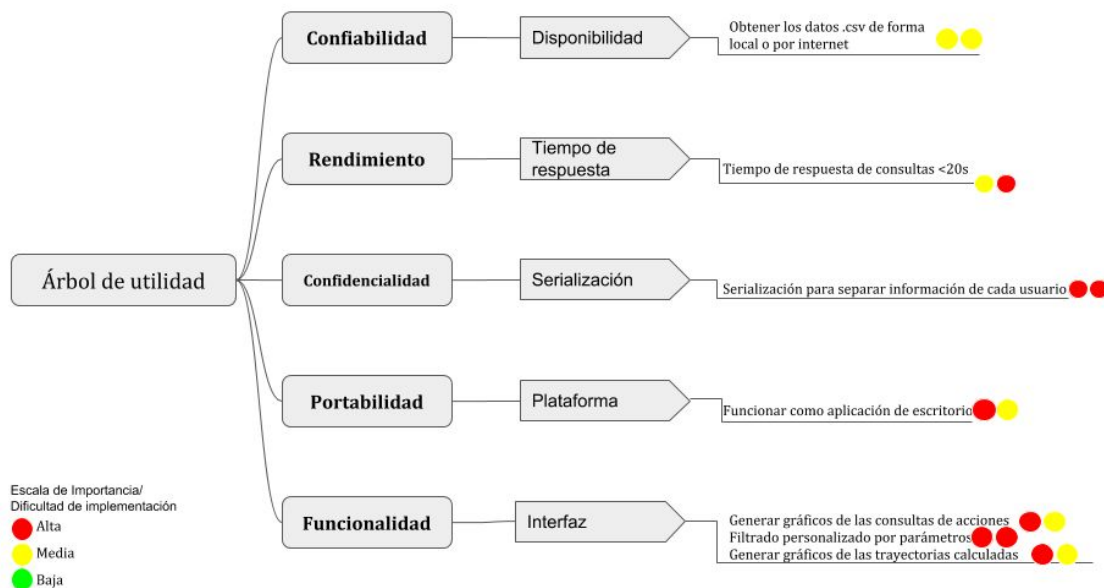


Figura 1: Árbol de utilidad del proyecto.

3. Modelo de Software (Final)

A continuación (Figura 2) se muestra el diagrama de clases del proyecto con las clases relevantes. Note que las clases con el tag <<component>>, a la hora de implementación, son clases que heredan de [React.Component](#) (por ello todas tienen el método render). Es de destacar que el modelo de Software es el mismo que el de la entrega anterior.

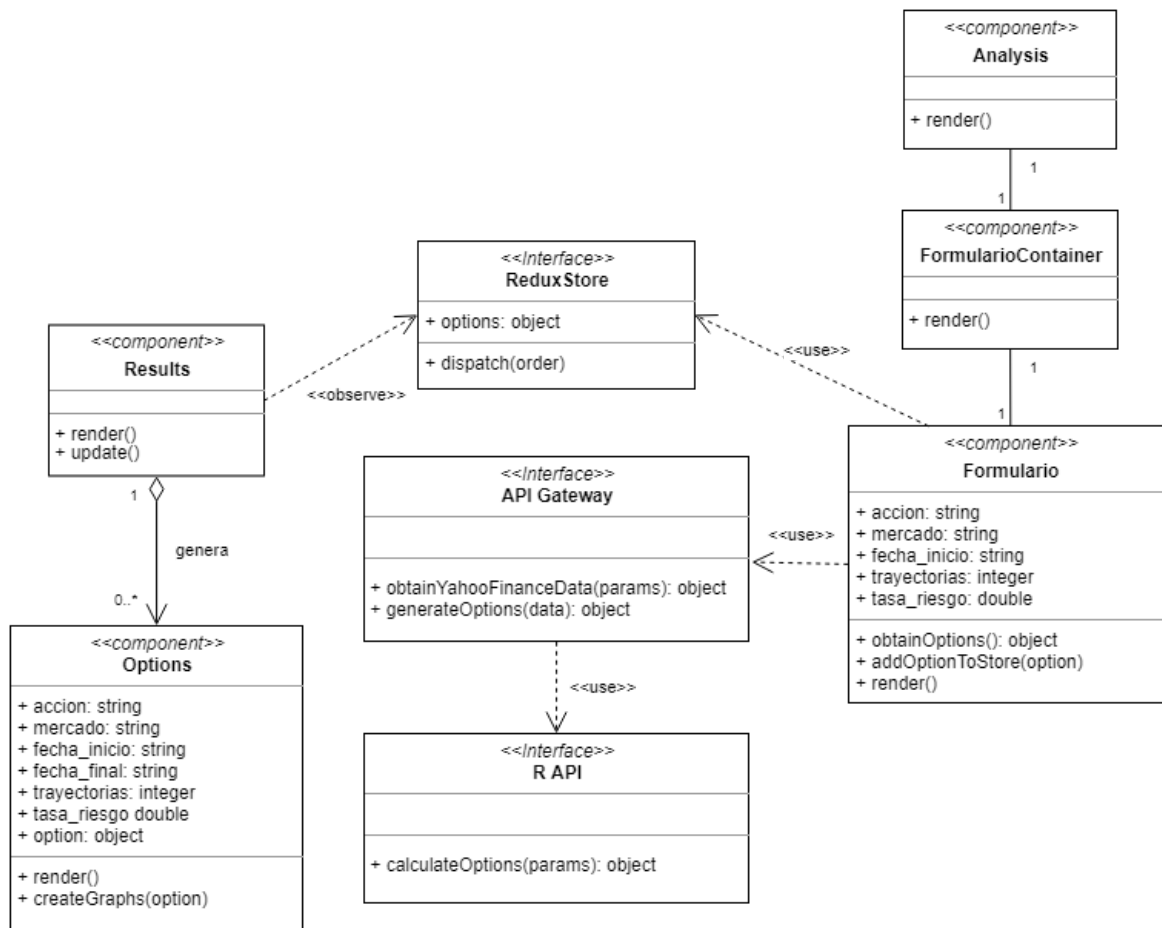


Figura 2: Diagrama de Clases del proyecto.

Además, en la tabla 3 se presenta la selección de patrones de diseño para el proyecto. Estos fueron escogidos en función de los requisitos del proyecto, junto con las herramientas escogidas que mejor los implementan.

Intención	Patrón de Diseño	Razonamiento
Generar historial de las consultas hechas por el consumidor	Observer	Se necesita actualizar el estado del componente que muestra los gráficos de las consultas hechas por el usuario (Results) a medida que se hagan solicitudes en el componente Formulario. La implementación de hacer aquello mediante comunicación directa entre los componentes involucrados en React implica una tediosa implementación y difícil debugeo, por lo que se dispondrá de una store que guarde los datos analizados, así Results, mediante la observación de la store, puede detectar cambios en los datos y luego actualizar automáticamente la vista.
Encapsular implementación de algoritmo de extracción de datos obtenidos por Yahoo! Finance, transformación y tratamiento de datos.	Gateway	Se necesita encapsular la interacción tanto con la API de Yahoo Finance como con la API de R dado que no se desea mezclar la lógica de muestra de datos de la vista con la lógica de extracción, transformación y tratamiento de datos. Para ello, se creó una API (que es el backend de la aplicación) que se encarga de la última lógica mencionada, así sólo basta que se haga consulta a esa API con los parámetros necesarios para entregar el resultado final.

Tabla 3: Patrones de diseño escogidos.

4. Trade-offs entre tecnologías (final)

El SIG (Softgoal Interdependency Graph) se presenta en la figura 3. Es de destacar que este no sufrió modificaciones ya que satisface los criterios establecidos.

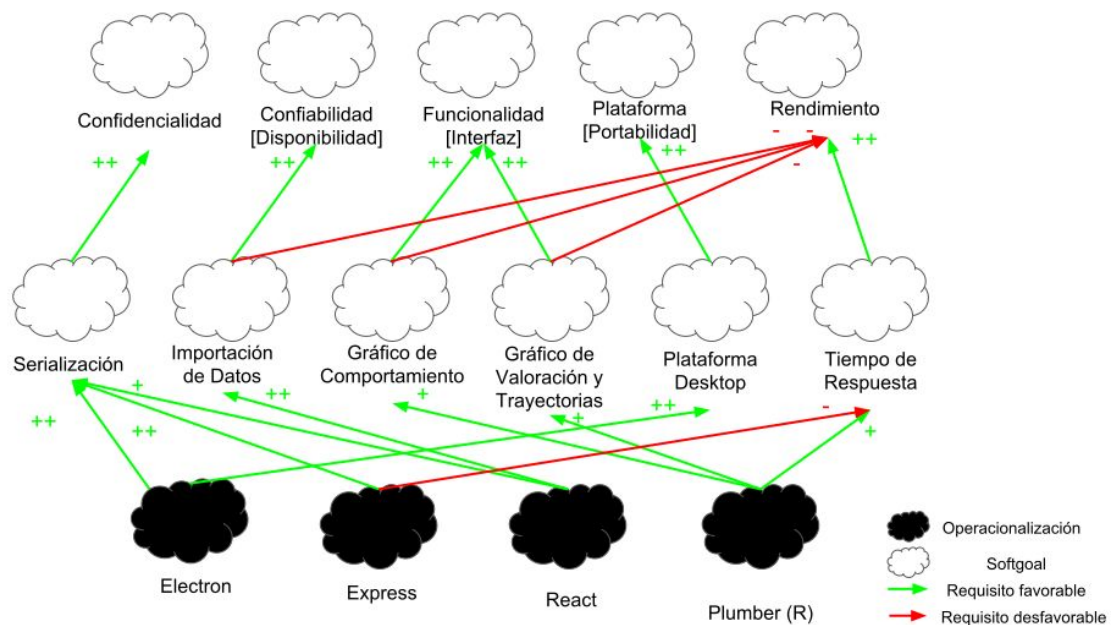


Figura 3: Softgoal Interdependency Graph (SIG).

Adicionalmente, los Trade-offs entre las opciones tecnológicas utilizadas se muestra en la tabla 4. De manera análoga al SIG, este no sufrió modificación alguna.

Decisión	Softgoal	Evaluación	Razonamiento
Serializar para separar la información de cada usuario	Confidencialidad	++	Al establecer un login de usuario mediante una “serial key” o contraseña asociada a cada usuario, se está resguardando la privacidad de los datos del mismo.
Serialización mediante Electron	Confidencialidad	++	Al establecer una “serial key” o contraseña de manera local (gestor de ventanas), se permitirá resguardar la información tanto online (datos de Yahoo Finance) como offline (simulaciones vía archivo propio del usuario).

Serialización mediante Express	Confidencialidad	++	Si no se tiene la llave al iniciar la aplicación, no se podrán realizar simulaciones.
Serialización mediante React	Confidencialidad	+	Si bien contribuye a la seguridad, poco ayudaría (y no tendría sentido) que el Front End (apariencia visual) de la aplicación maneje dicha acción.
Importación de datos mediante archivo (tanto online como offline)	Confiabilidad (Disponibilidad)	++	Si no se tiene conexión a internet, es posible realizar simulaciones con archivos descargados que contengan la información sobre las opciones.
Importación de datos vía React	Confiabilidad (Disponibilidad)	++	Permite el uso de una interfaz de usuario sencilla para el usuario que quiera cargar su archivo .csv para analizar.
Importación de datos	Rendimiento	-	Poco ayudaría considerando lo extenso, largo o “pesado” que pueden llegar a ser los archivos que cargue el usuario.
Gráfico de Comportamiento ayuda a tener una UI más amigable	Funcionalidad (Interfaz)	++	El tener el gráfico en la interfaz de usuario permite ver que el programa, y su simulación, funcionan correctamente. Además, es útil tener dicha información de inmediato.
Gráfico de Valoración y Trayectorias ayuda a tener una UI más amigable	Funcionalidad (Interfaz)	++	Mismo razonamiento anterior.

Gráfico de comportamiento	Rendimiento	-	Poco ayudaría el tener un gráfico con los resultados en cuanto a tiempos de carga del mismo en la aplicación y/o en la obtención el mismo vía la API de R o generación vía JavaScript.
Gráfico de Valoración y Trayectorias	Rendimiento	-	Mismo razonamiento anterior.
Uso de Plumber (R) para generar gráficas	Funcionalidad (Interfaz)	++	Si las gráficas se generan al llamar a la API de R (después de simular), sería bastante rápido ocupar los mismos y solamente mostrarlos por pantalla.
Portar la aplicación a formato “Desktop” mediante Electron	Plataforma (Portabilidad)	++	Al ocupar un framework (set de herramientas) que permite portar la aplicación a Windows, Mac o Linux, simplifica bastante la programación al no tener que desarrollar código para cada sistema operativo.
Despliegue de resultados vía Electron	Rendimiento	-	Si los servicios utilizados demoran en responder de vuelta, la aplicación demorará también en responder.
Uso de Plumber (R) para analizar datos	Rendimiento	++	El cliente mismos decidió que todos los cálculos se realicen utilizando R debido a su precisión numérica y rapidez.

Tabla 3: Trade-offs entre opciones tecnológicas.

5. Deuda técnica incurrida

Considerando que la **deuda técnica** son decisiones tomadas por el equipo para lograr cumplir los plazos de entrega, **sabiendo** que estas podrían crear problemas de mantenibilidad a futuro, estas se presentan en la tabla 4

Ítem deuda técnica	Razonamiento	Impacto
Implementación de funciones de menú “Alt”	Menú desplegable clásico de las aplicaciones de escritorio no fue completamente implementado	Todas las funcionalidades del menú “alt” no fueron implementadas, lo que pudo haber resultado en una mejor experiencia de uso para el cliente en primera instancia, implementarlas hubiese aumentado la usabilidad, en tanto crea una opción para explorar las funcionalidades de la aplicación, para quienes se sientan más cómodos con ese formato.
Acceso no restringido a web servers implementados (React y Express).	Rápida implementación para probar funcionamiento, robustez y compatibilidad de tecnologías con la aplicación a desarrollar.	Permite acceso no autorizado a los web servers utilizados para la implementación de la aplicación (cualquier que sepa donde se “ubican” los servidores web y sus puertos, puede acceder a ellos y ocupar la aplicación desde un entorno para el cual no fue desarrollada y, además, explotar las capacidades de los mismos mediante consultas repetidas a ellos, más aún si no se cuenta con tokens de autenticación).
Pruebas de archivos locales distintos a los de Yahoo finance	Inspección de archivos .csv que contengan datos sobre las acciones que no sigan el mismo formato que los descargados y probados de Yahoo Finance.	Acondicionar el sistema para ser tolerante a pequeños o grandes cambios en el formato de los arreglos que están escritos en los archivos .csv, ya que comparten ciertas reglas que habría que definir en el

		software. Esto agregaría valor al software en el sentido de asegurar al usuario la lectura de sus archivos sin condicionar la fuente
--	--	--

Tabla 4: Deuda técnica del proyecto.

