



Departamento de Informática
Universidad Técnica Federico Santa María



Informe de Proyecto – INF-225-2018-1-CC
Proyecto “WhoMe”
29-08-2018

Integrantes:

Nombres y Apellidos	Email	ROL USM
Eduardo Arancibia	eduardo.arancibia.13@sansano.usm.cl	201330011-1
Luis Araya	luis,araya.13@sansano.usm.cl	201330045-6
Felipe Condon	felipe.condon.13@sansano.usm.cl	201303018-1

1. Requisitos clave	3
2. Árbol de Utilidad	4
3. Modelo de Software	5
4. Trade-offs entre tecnologías	6

1. Requisitos clave

Req. funcional	Descripción y medición
Seleccionar opciones	El sistema debe permitir seleccionar que tipo de opción de compra se usará (cada opción realiza distintos cálculos)
Generar trayectorias	El sistema debe permitir ingresar la cantidad de trayectorias a generar para la simulación
Importar datos	El sistema debe permitir importar datos desde la API de Yahoo Finance
Subir datos offline	El sistema debe permitir el ingreso de datos proveídos por el usuario
Simulación	El sistema debe usar el método de Montecarlo para el análisis estadístico
Seleccionar fechas de entradas	Ingresar parámetros de fechas para realizar la simulación

Tabla 1: Requisitos Claves funcionales

Req. extra-funcional	Descripción y medición
Desempeño	Se debe usar lenguaje R para el cálculo estadístico.
Gráficos acciones	Se debe mostrar un gráfico con el valor de la acción en función del tiempo
Entorno de escritorio	El sistema debe ejecutarse en un ambiente Dekstop y sin la necesidad de conexión a internet

Tabla 2: Requisitos Claves extra funcionales

2. Árbol de Utilidad

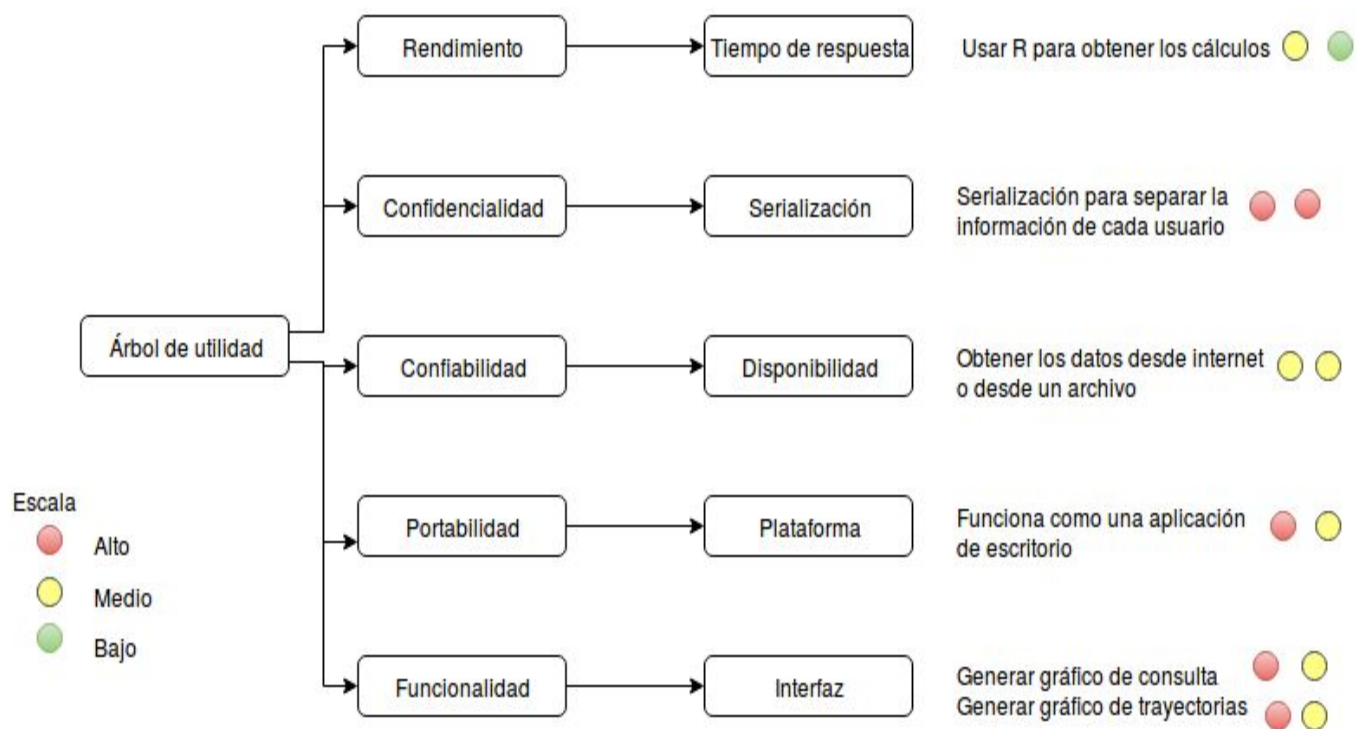


Figura 1: Árbol de Utilidad.

3. Modelo de Software

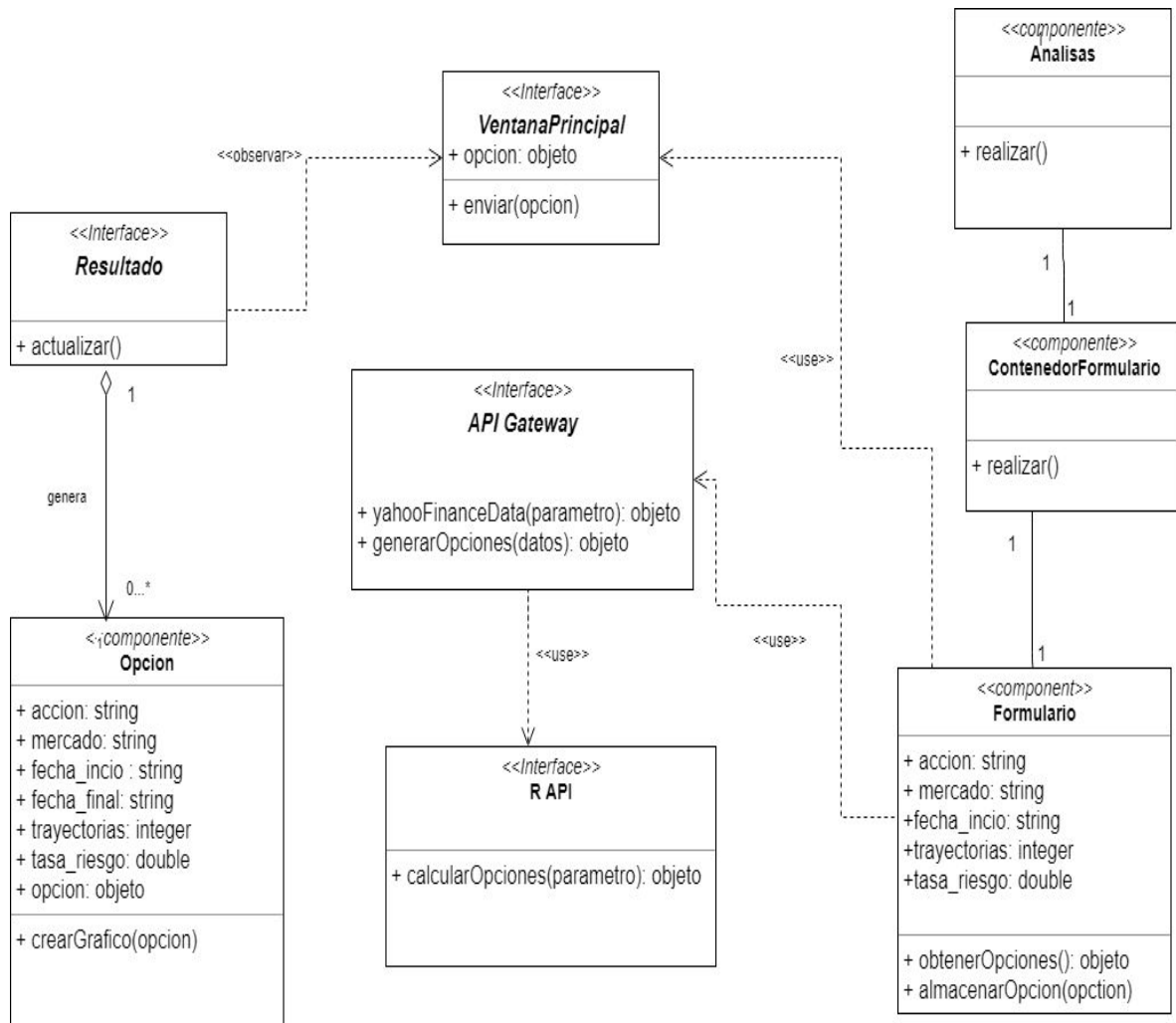


Figura 2: Diagrama de clases del proyecto.

Intención	Patrón de Diseño	Razonamiento
Al momento de implementar la extracción de datos obtenidos por Yahoo Finance se encapsula este algoritmo, además de la transformación y tratamiento de los datos,	Gateway	Se debe encapsular la API de Yahoo Finance y de R, ya que, no se quiere mezclar la lógica de la obtención de datos, transformación y tratamiento de datos. Para evitar esto, existe una API, el cual cumple la función de backend, que se procura del tratamiento de datos, siendo necesario solo realizar consultas a la API con los parámetros adecuados para obtener el resultado final.
Generar historial de las consultas hechas por el cliente	Observer	Al momento de actualizar los componentes, que los gráficos obtenidos (Resultado), al crear una solicitud al componente Formulario, al realizar una comunicación directa, se hace un tanto complicado, por lo que se irán almacenando los datos analizados, así Resultado, por medio de la VentanaPrincipal, detecta modificaciones en los datos analizando automáticamente la vista.

Tabla 3: Patrones utilizados

4. Trade-offs entre tecnologías

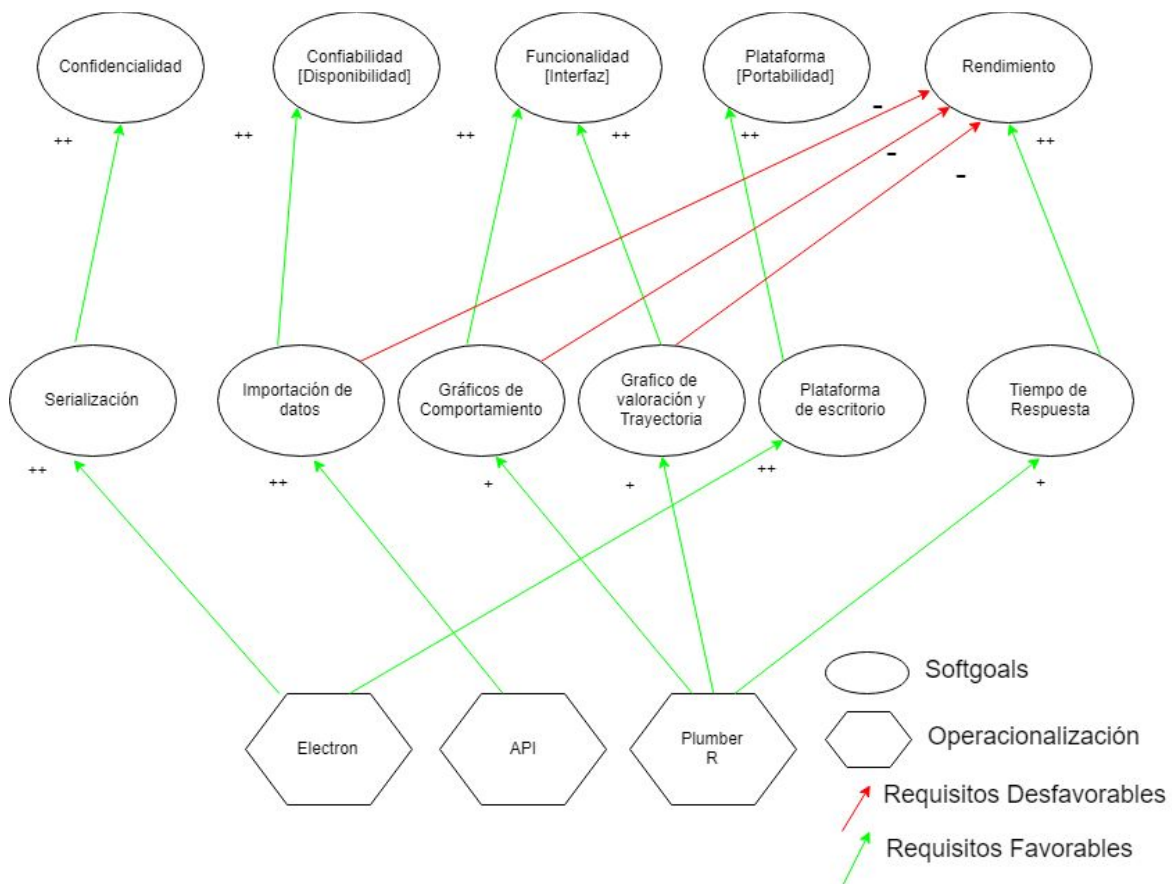


Figura 3: Softgoal Interdependency Graph.

Decisión	Softgoal	Evaluación	Razonamiento
Uso de Plumber (R) para analizar datos	Rendimiento	++	El cliente mismo decidió que todos los cálculos se realicen utilizando R debido a su precisión numérica y rapidez.
Despliegue de resultados vía Electron	Rendimiento	-	Si los servicios utilizados demoran en responder de vuelta, la aplicación también demora en responder.
Portar la aplicación a formato "Desktop" mediante Electron	Plataforma (Portabilidad)	++	Al ocupar un framework que permite portar la aplicación a Windows, Mac o Linux, simplifica bastante la programación al no tener que desarrollar código para cada sistema operativo.
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.
Gráfico de Valoración y Trayectorias	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 comportamiento	Rendimiento	-	Mismo motivo anterior.
Gráfico de Valoración y Trayectorias ayuda a tener una UI más amigable	Funcionalidad (Interfaz)	++	Mismo razonamiento anterior.

Gráfico de Comportamiento ayuda a tener una UI más amigable	Funcionalidad (Interfaz)	++	El tener el gráfico en la UI permite al usuario ver que el programa, y su simulación, funcionan correctamente. Además, es útil tener dicha información de inmediato.
Importación de datos	Rendimiento	-	Poco podría ayudar al rendimiento considerando lo extenso, largo o “pesado” que pueden llegar a ser los archivos que cargue el usuario.
Importación de datos mediante archivo (tanto online como offline)	Confiabilidad (Disponibilidad)	++	Si no se tiene conexión a internet, es posible realizar las simulaciones consultando la API de R (offline).
Serialización mediante Electron	Confidencialidad	++	Al establecer un “serial key” mediante Electron, y si no se tiene el correcto, la aplicación no seguirá el curso normal (no se genera la ventana de carga de datos).

Tabla 4: Trade-offs.

5. Deuda Técnica

Ítem deuda técnica	Razonamiento	Impacto
Demora en la simulación	Se usa el método montecarlo para simular, el tiempo en hacer la simulación depende de las iteraciones.	Afecta en especial el rendimiento del sistema, además, si se aumentan las trayectorias a generar, el tiempo aumenta
Uso de Electron para la interfaz	Permite portar fácilmente la aplicación a Windows, Linux, Mac o Android	Pueden encontrar la interfaz "fea"
API Yahoo finance	Fácil de usar y requerido por el cliente	Puede ser inseguro al no establecer mayores medidas de seguridad

Tabla 5: Deuda Técnica