

¿Dónde invierto?

Segunda Parte (Segundo Cuatrimestre)

1 Contexto	1
2 Entregas	1
2.1 Cuarta Entrega - Persistencia	1
2.2 Quinta Entrega - UI Web	2
2.3 Sexta Entrega - Cambios Arquitecturales	2

1 Contexto

A la vueltas de las vacaciones de Sabrina, Héctor le cuenta que en estas cuatro semanas las cosas cambiaron mucho.

Primero, porque ahora el mismo Héctor tiene serias dudas sobre su idea original: llegó a la conclusión de que nada es tan fácil como parece, que el mundo de la bolsa es demasiado competitivo y aún hay aspectos que no domina como para poder dar asesoramiento a más personas.

Segundo, porque charlando con un amigo, éste opinó que de todas formas el software que habían creado junto a Sabrina era una herramienta interesante para pequeños inversores, y que él consideraba que muchos estarían dispuestos a pagar por éste.

Todo esto lo llevó a Héctor a plantear un nuevo modelo de negocio para su incipiente emprendimiento: en lugar de asesorar a clientes, proveer directamente un software para quienes sí lo hacen. Y tras varias reuniones más llegó a Valeria, empresaria que está dispuesta a ayudar económicamente a desarrollar este nuevo producto.

Por teléfono, Héctor le acaba de preguntar a Sabrina si ella quiere ser la arquitecta de un nuevo sistema, esta vez Web y con soporte para múltiples usuarios, basándose en el anterior. Y tras pensarlo un poco, no sólo Sabrina dice que está dispuesta a hacerlo, sino que además, considera que puede reutilizar mucho del viejo sistema.

2 Entregas

2.1 Cuarta Entrega - Persistencia

Hasta el momento persistíamos los datos en archivos. El objetivo de esta entrega consiste en adaptar esta arquitectura para centralizar el almacenamiento. Es decir, necesitamos permitir almacenar los elementos del dominio en un medio relacional, no siendo necesario mantener la persistencia en archivos. La única excepción serán las cuentas, que por ahora seguiremos cargando tanto desde la base de datos como desde un archivo.

Alternativas: algunos motores de bases de datos recomendados son los siguientes:

- MySQL
- PostgreSQL

Dado que son de código abierto, maduros y cuentan con excelente soporte a lo largo de una gran variedad de lenguajes, frameworks y bibliotecas. Otros motores maduros pero de código propietario son:

- SQLServer
- Oracle

Además, no queremos acoplarnos a una base de datos particular, y para poder reutilizar el modelo de objetos ya construido, se pide el uso de un ORM. Si bien parece fácil, ¡cuidado!, puede que haya que realizar modificaciones al modelo original.

Sugerencia: De utilizar Java 8 o posterior, recomendamos utilizar JPA/Hibernate (es decir, Hibernate a través de la interfaz estándar de JPA). Un buen punto de partida para adaptar el proyecto es el siguiente: <https://github.com/dds-utn/jpa-proof-of-concept-template>.

Se pide:

1. Introducir en la solución un motor de base de datos relacional
2. Introducir en la solución un motor ORM
3. Persistir en dicha base de datos las siguientes entidades:
 - a. Cuentas
 - b. Indicadores
 - c. Metodologías
 - d. Empresas
4. Lograr que la aplicación siga funcionando sin cambios funcionales
5. Lograr que la aplicación pueda seguir cargando cuentas desde archivos. No es necesario desarrollar nuevas pantallas, sino sólo continuar soportando esta carga programáticamente.
6. Testear adecuadamente la persistencia.