



Pontificia Universidad Católica de Chile

Escuela de Ingeniería Civil

Departamento de Ciencias de la Computación

Ingeniería de Software

Ingenieros de Software Grupo 11

Alvaro Alemparte - Matías Hurtado - Benjamín Ibarra

12 de Octubre de 2015

Elección del modelo a utilizar

El primer modelo tiene la ventaja de ser una representación y solución más simple del problema presentado y así, determina una implementación más rápida, pero no quedaría muy sujeta al problema inicial y no sería muy escalable.

Además, posee alto acoplamiento, pues se recurre constantemente a una misma clase para acceder a atributos o métodos que podrían estar en clases separados o heredadas. Se dificulta la modularidad y reutilización del código, pues no es posible extender o modificar fácilmente el código al estar tan concentrado. También tiene baja cohesión, dado que hay mucha información que no está realmente relacionada en una misma clase, por ejemplo: si una persona es cliente, no necesita el atributo sueldo.

El segundo modelo tiene la ventaja de presentar tener **alta cohesión**. Esto se debe a que nos hemos preocupado de asignar correctamente las responsabilidades a cada uno de los entes. También cumple el principio de **bajo acoplamiento**, de modo que se favorece la modularidad y la reusabilidad.

Por las razones anteriores, es que decidimos utilizar el segundo modelo como referencia para el desarrollo del proyecto, pues es bastante más completo y coherente con los principios de diseño estudiados.

Cambios en el modelo utilizado

Pese a que elegimos el segundo modelo como referencia, a medida que construimos la programación, nos dimos cuenta de que algunas cosas importantes faltaban. A continuación se exponen los cambios o clases agregadas y la importancia de cada una.

- Se agregó una clase **Fachada**, llamada **Sistema**, que se encargará de coordinar las demás clases, conectarlas y distribuir responsabilidades, de modo de facilitar la comunicación cuando se implemente la interfaz gráfica. Además de utilizar el principio de fachada, se incluyó el patrón **Singleton**, para que esta fachada sea única y esté disponible desde cualquier otra instancia del programa.



Pontificia Universidad Católica de Chile

Escuela de Ingeniería Civil

Departamento de Ciencias de la Computación

Ingeniería de Software

Esta clase también será la responsable de conservar los datos activos guardados en el computador, de modo que leerá los archivos guardados al momento de iniciar y guardará al momento de cerrar.

- Se conecta la clase Mensajes con la clase Sucursal, pues los mensajes son enviados hacia una sucursal ingresando a una cola de espera.