



Patró Adaptador

Estem dissenyant una classe que necessita d'un servei, que ja hem implementat anteriorment i que volem reutilitzar. Però la signatura dels mètodes que volem reutilitzar no s'adapten exactament als requeriments de la nova classe. En aquests casos podem:

1. Modificar la classe que proporciona el servei:
 - Modificant els mètodes per adaptar-los a les noves necessitats, o
 - Creant nous mètodes per proporcionar el servei que necessita la nova classe.
2. O bé: Modificar la nova classe, de forma que s'adapti als requeriments del servei ja implementat.
3. O bé: Crear un adaptador, que faci de pont entre la funcionalitat esperada per la nova classe, i la que proporciona el servei ja implementat.

La primera opció proposa la modificació del servei. En general, un servei s'implementa amb la intenció de proporcionar una funcionalitat general, que pot ser aprofitada per diverses classes. Atenent a això, no és aconsellable modificar un servei per adaptar-lo a les necessitats concretes d'una classe. Tot i que és cert que moltes vegades un servei s'implementa en origen per satisfer unes necessitats concretes, si aquestes necessitats són prou genèriques, és convenient no modificar-lo. Per altra banda, el fet de crear nous mètodes al servei que satisfacin les necessitats concretes de la nova classe, no és més que una altra forma d'adaptar el servei a la classe; i és el que tractem d'evitar.

Això ens porta al punt 2: Modificar la nova classe i adaptar-la a la funcionalitat proposada pel servei. Aquesta tampoc és una bona opció, perquè les necessitats de la nova classe són les que són, i no poden estar coartades per un conjunt de funcionalitats genèriques, que són les que proposa el servei. Per tat, hem de trobar una altra solució.

El punt 3 proposa el següent: Les necessitats de la nova classe són les que són, com ja hem dit. I la funcionalitat exposada pel servei és genèrica i, per tant, rarament podrà haver un aprofitament directe. Es necessari un pont, que serveixi de comunicació entre el servei i la nova classe, i que adapti la necessitat de la nova classe amb la funcionalitat genèrica proposada pel servei.

Característiques

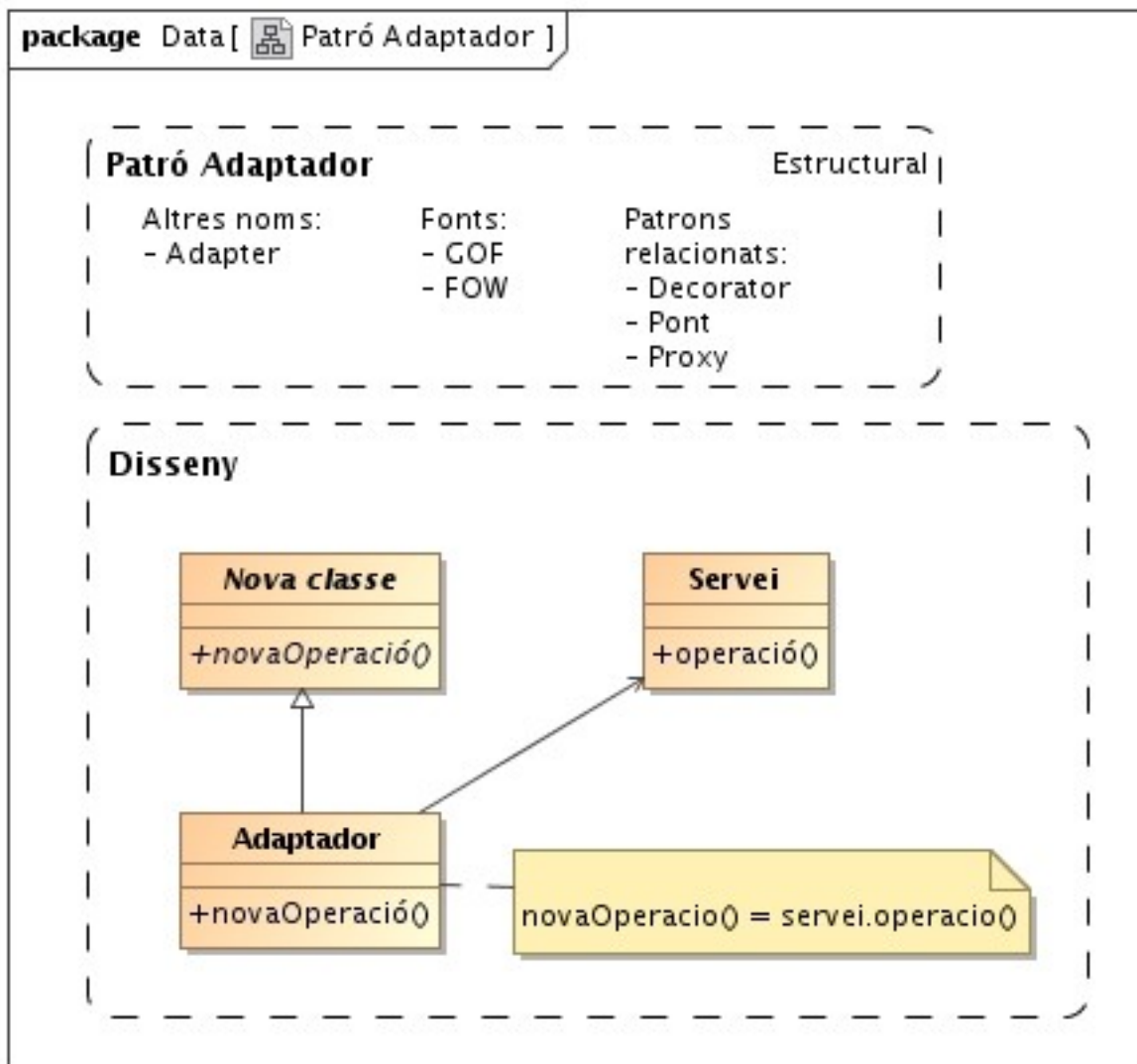
| Nom del patró | Tipus de patró | Fonts documentals |
|---------------|--------------------------------------|-------------------|
| Adaptador | Estructural | GOF FOW |
| Altres noms | Patrons relacionats | |
| Adapter | Decorador Pont, (Bridge) Proxy | |



El patró **Adapter** proposa una interfície diferent per al servei real al qual adapta, mentre que **Proxy** implementa en la seva interfície els mateixos mètodes que el servei real, (o el subconjunt útil d'aquests mètodes per la classe Subjecte). Proxy es transparent en el seu ús al servei al que substitueix, si no és que opcionalment pot implementar mètodes per a garantir la seguretat, o establir controls; mentre que Adapter proposa canvis de funcionament respecte al servei al que adapta.

El patró **Decorator** afegeix noves responsabilitats, (proporciona nova funcionalitat emprant la herència), sobre una classe concreta. No representa un servei.

Disseny





Nova classe

Aquesta classe és un client, que implementa una sèrie de mètodes que poden fer ús, (mitjançant un adaptador), d'un servei.

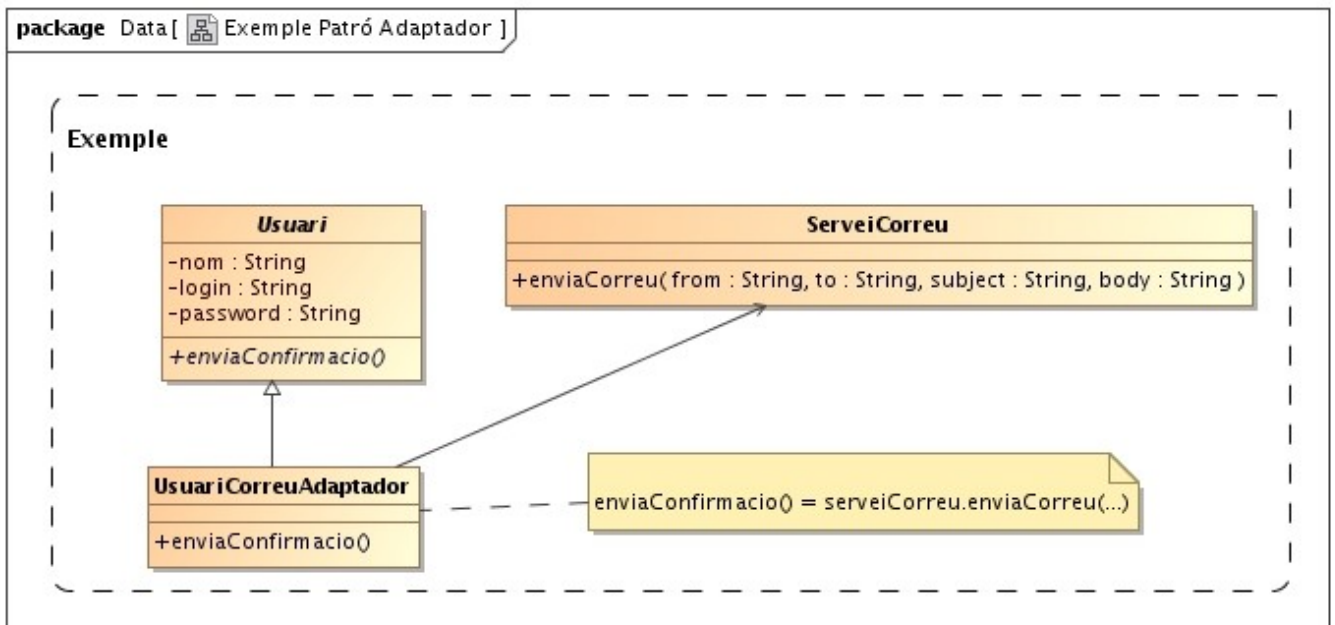
Adaptador

Aquesta classe "adapta" la necessitat de la nova classe amb la funcionalitat proporcionada pel servei.

Servei

Classe que implementa la funcionalitat genèrica.

Exemple



En l'exemple tenim una aplicació que gestiona les dades d'un usuari. Una de les operacions de l'usuari és l'enviament d'un correu electrònic. Aquest enviament es fa mitjançant l'ús d'un servei ja existent, (**ServeiCorreu**). Com que la implementació d'aquest servei és de característiques diferents a les que necessita **Usuari**, creem un adaptador, (**AdaptadorCorreuUsuari**), que "adapta" la petició de l'usuari amb els mètodes genèrics del servei.