



Cenni Object Design Document

MinervHub



Riferimento	NC07_CenniODD
Versione	2.0
Data	22/12/2025
Destinatario	Docente di Ingegneria del Software 2025/26
Presentato da	NC07 - Gennaro Pio Albano (G.P.A.), Giuseppe Annunziata (G.A.), Alessandro Bonelli (A.B.)
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
24/11/2025	1.0	Prima stesura	G.P.A.
24/11/2025	1.1	Proxy pattern	A.B.
24/11/2025	1.2	Facade pattern	G.A.
22/12/2025	2.0	Revisione finale ed ultime modifiche	Tutto il team

Indice

Design Pattern.....	3
Proxy.....	3
Facade.....	4

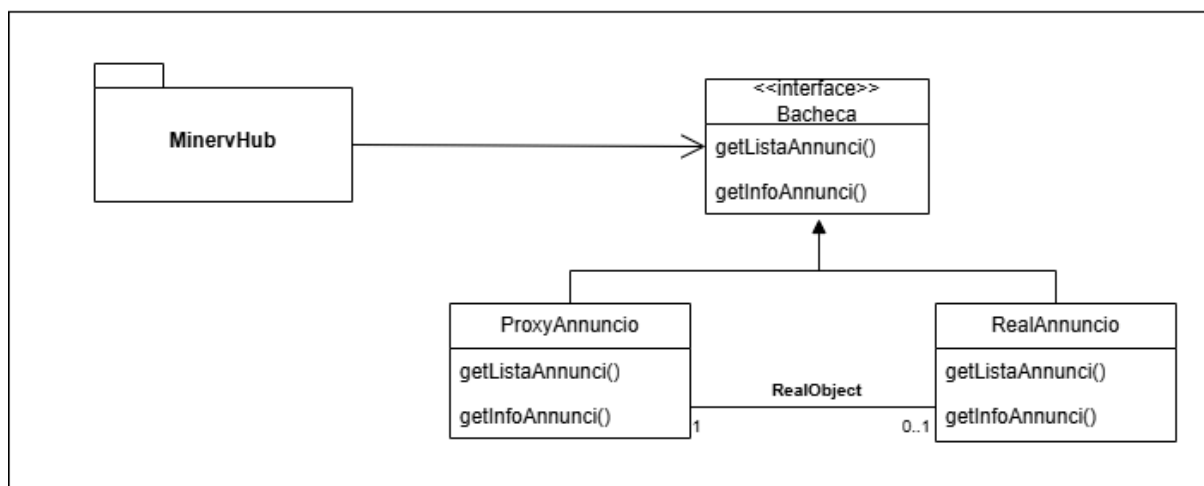
Design Pattern

Per ogni pattern viene descritto lo scenario applicativo, con una breve spiegazione del problema che risolve e del contesto d'uso. Inoltre, viene fornita una descrizione del funzionamento del pattern e un diagramma delle classi coinvolte, evidenziando come il pattern viene integrato nell'architettura complessiva del sistema.

Proxy

Il design pattern Proxy è un design pattern strutturale che consente di ritardare la creazione e il caricamento completo di oggetti, fino al momento in cui sono effettivamente necessari. Questo approccio migliora l'efficienza, riducendo l'uso delle risorse di sistema e ottimizzando il caricamento dei dati. Nel nostro caso, potremmo utilizzare il Proxy nel sottosistema Bacheca, nello specifico per la gestione della lista degli annunci pubblicati: Quando l'utente accede alla sezione bacheca, viene mostrato un elenco degli annunci con informazioni di base (ad esempio, titolo dell'annuncio, corso, tariffa). In questo momento non è necessario caricare tutti i dettagli degli annunci. Per questo motivo, utilizziamo un oggetto proxy che rappresenta ogni annuncio nella lista e fornisce solo i dati essenziali. Quando l'utente seleziona un annuncio specifico, il Proxy richiede e carica i dettagli completi attraverso l'oggetto reale. In questo modo, il caricamento dei dettagli viene effettuato solo per gli annunci richiesti dall'utente.

Proxy Pattern



Facade

Il design pattern Facade è un design pattern strutturale che fornisce un'interfaccia semplificata e unificata per accedere a un sottosistema complesso, nascondendo i dettagli implementativi e le interazioni interne. Questa soluzione è particolarmente utile nei sistemi con un'architettura chiusa, dove si desidera limitare l'interazione diretta con le complessità interne. Nel nostro caso, utilizziamo il Facade Pattern nello Strato di Logica Applicativa, nello specifico implementandolo attraverso l'interfaccia di AnnuncioService. Quando un Controller dello strato Web riceve una richiesta, come ad esempio la creazione di un nuovo annuncio, si trova a dover gestire diverse operazioni complesse. In questo momento non è necessario che il Controller conosca o coordini direttamente la validazione dei dati, le regole di business o le chiamate al database.

Facade Pattern

