

MediaLog

Ingegneria del Software L-A

Fabbrica Simone

Morlini Gabriele

Grandi Andrea

Indice generale

MediaLog.....	3
Introduzione.....	3
Documento dei requisiti.....	4
Scansione di un nuovo supporto.....	4
Ricerca di un file.....	4
Gestione prestito supporti.....	4
Glossario.....	5
Casi d'uso.....	6
Scenari.....	8
SCENARIO 1.0 - Aggiunta supporto al catalogo.....	8
SCENARIO 2.0 - Ricerca:.....	8
SCENARIO 3.0 - Impostazione chiave di ricerca:.....	9
SCENARIO 4.0 - Inserimento di un nuovo prestito:.....	9
Modello statico.....	10
Modello Dinamico.....	11
Progettazione.....	12
Note.....	12
Diagrammi UML.....	13
Pattern.....	17
Composite.....	17
Strategy.....	19
Factory.....	19
Principi di progettazione.....	20
Dependency Inversion Principle (DIP).....	20

MediaLog



MediaLog permette di catalogare il contenuto di supporti di archiviazione come CD, DVD o Blu-Ray.

Introduzione

Solitamente il backup dei dati viene eseguito su CD e DVD nei quali però risulta difficile tenere traccia del contenuto. La ricerca dei dati archiviati, qualora non si ricordi esattamente dove sono memorizzati, può diventare un'operazione estremamente lunga e noiosa. Allo stesso tempo può tornare molto comodo visualizzare la lista dei file contenuti in un supporto senza che questo debba essere fisicamente inserito nel PC.

Inoltre può essere utile gestire in modo semplice il prestito dei propri CD agli amici.

MediaLog si propone come soluzione a questo tipo di problemi, catalogando automaticamente il contenuto di CD e DVD e consentendo di effettuare rapide ricerche.

Documento dei requisiti

All'avvio del programma l'utente deve avere accesso alle seguenti funzioni:

Scansione di un nuovo supporto

Questa operazione consente all'utente di inserire nel catalogo un nuovo supporto.

L'utente inserisce un supporto di un determinato tipo (CD, DVD o Blu-Ray) nel computer ed il sistema, effettuando automaticamente la scansione, memorizza sul catalogo il contenuto di quest'ultimo. Ogni supporto ha un filesystem strutturato in direttori e file: un direttorio può contenere altri direttori e/o file. Per ogni direttorio e file si tiene traccia del nome; per ogni file si tiene traccia anche della dimensione e della propria estensione.

Il sistema è pensato per dispositivi di sola lettura.

Nel caso in cui il supporto si danneggi o sia smarrito, deve essere possibile effettuarne la cancellazione dal catalogo.

Il sistema al termine della scansione fornisce all'utente un identificativo numerico univoco progressivo che può essere trascritto sul dispositivo per facilitarne il ritrovamento in futuro.

Ricerca di un file

Se sono presenti dei supporti nel catalogo, l'utente può eseguire la ricerca di un file per vedere in quali supporti è eventualmente contenuto.

L'utente che necessita di reperire un file, effettua la ricerca sull'intero catalogo e rapidamente il sistema comunica la lista di supporti che lo contengono e le informazioni relative ad essi. La chiave di ricerca può essere data dal nome del file, dall'estensione oppure dalla dimensione. L'utente può inoltre decidere di combinare arbitrariamente le chiavi di ricerca disponibili.

Il Sistema consente inoltre, dato l'identificativo di un supporto, di visualizzarne il contenuto.

Gestione prestito supporti

Il Sistema prevede infine la gestione del prestito dei supporti presenti nel catalogo. In particolare è presente un'anagrafe delle persone conosciute di cui si tiene traccia del nome, cognome, telefono, e-mail e di un nickname univoco. Al momento del prestito l'utente inserirà nel sistema l'identificativo del supporto e il nickname della persona e il sistema provvederà a memorizzarli insieme alla data corrente.

Per ogni prestito è possibile visualizzare la data in cui è stato effettuato. Non è prevista la gestione delle scadenze dei prestiti: il sistema si limita solamente a fornire la lista dei prestiti effettuati e consente di gestire la restituzione.

Inoltre per ogni supporto deve essere possibile visualizzare le informazioni sulla persona a cui è stato eventualmente prestato.

Glossario

Catalogo: elenco contenente tutti i supporti di cui l'utente vuole tenere traccia.

Supporto (di archiviazione): dispositivo su cui si registrano informazioni (dati). E' caratterizzato da:

- Identificativo (numerico): nome univoco associato ad ogni supporto.
- Tipo: ad esempio CD, DVD o Blu-Ray

Dispositivo di sola lettura: un qualunque Supporto di archiviazione il cui contenuto non può cambiare.

Contenuto (di un supporto): è un insieme di file e direttori memorizzati sul supporto.

File: sequenza (di byte), immagazzinati come un solo elemento su un supporto. E' caratterizzato da:

- Nome
- Dimensione
- Estensione.

Direttorio: elemento del filesystem utilizzato per ordinare i file in una gerarchia. E' caratterizzato da un nome.

Utente: attore principale del sistema MediaLog.

Scansione: permette di memorizzare il contenuto di un supporto nel catalogo.

Risultato ricerca: nomi dei supporti che contengono almeno un file corrispondente alla chiave di ricerca.

Chiave di ricerca: vincolo in base al quale viene effettuata la ricerca. Può essere:

- Il nome del file
- L'estensione
- La dimensione, intesa come intervallo tra un valore massimo e un valore minimo

Criterio di ricerca: vedi Chiave di ricerca

Prestito: temporanea concessione di un Supporto ad una Persona. Deve essere memorizzata la data di inizio del prestito.

Anagrafe: elenco di tutte le persone a cui l'utente può prestare un supporto.

Persona: destinataria del Prestito, caratterizzata da:

- Nickname univoco all'interno del sistema
- Nome
- Cognome
- Numero di telefono
- Indirizzo e-mail

Lista dei Prestiti: elenco di tutti i Prestiti di cui l'utente vuole tener traccia.

Casi d'uso

Il sistema prevede un solo attore: l'Utente.

Sono stati individuati 5 casi d'uso principali: Gestione catalogo, Ricerca, Visualizza contenuto supporto, Gestione anagrafe e Presta.

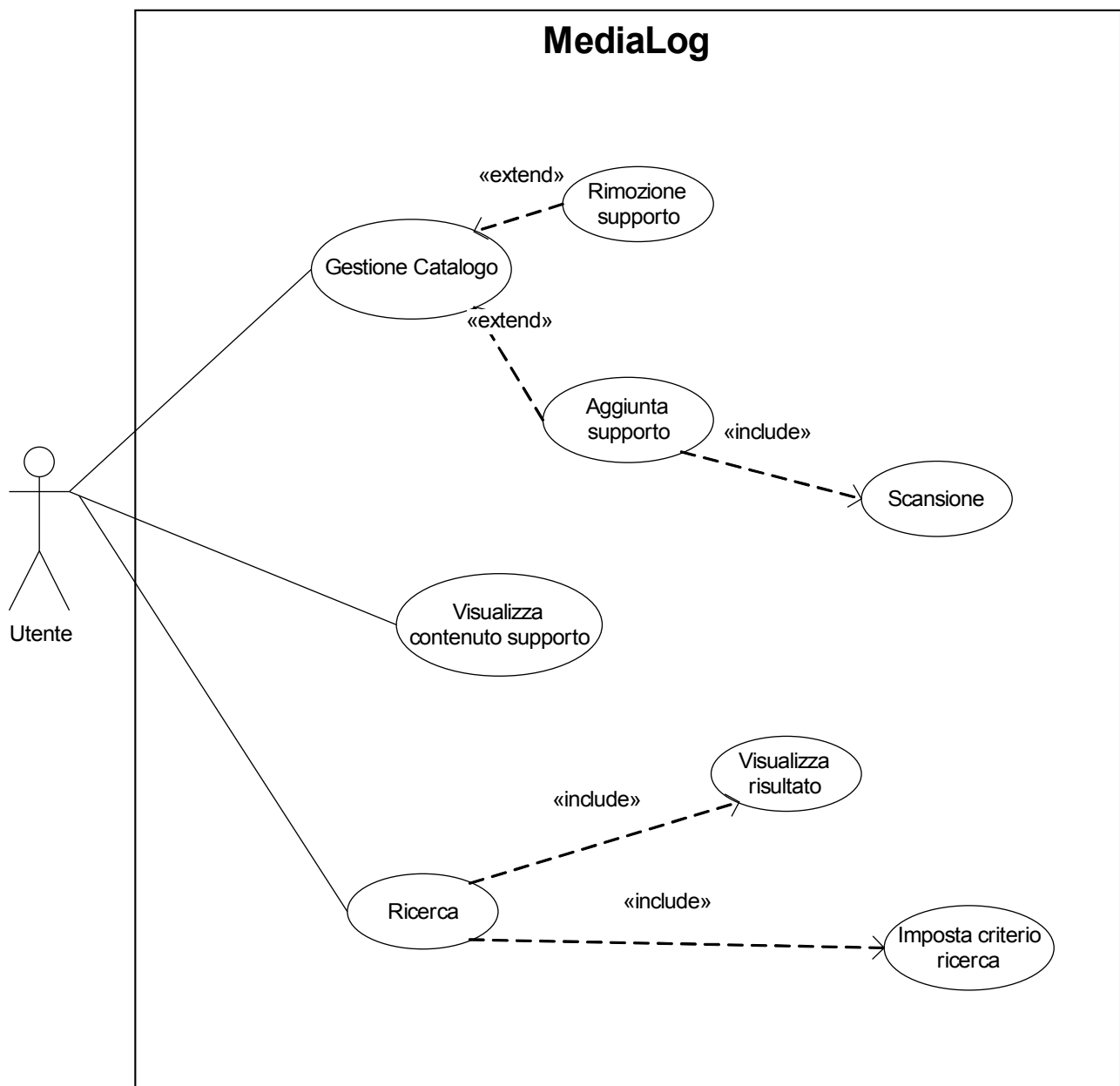


fig. 1: casi d'uso Catalogo / Ricerca / Visualizza supporto

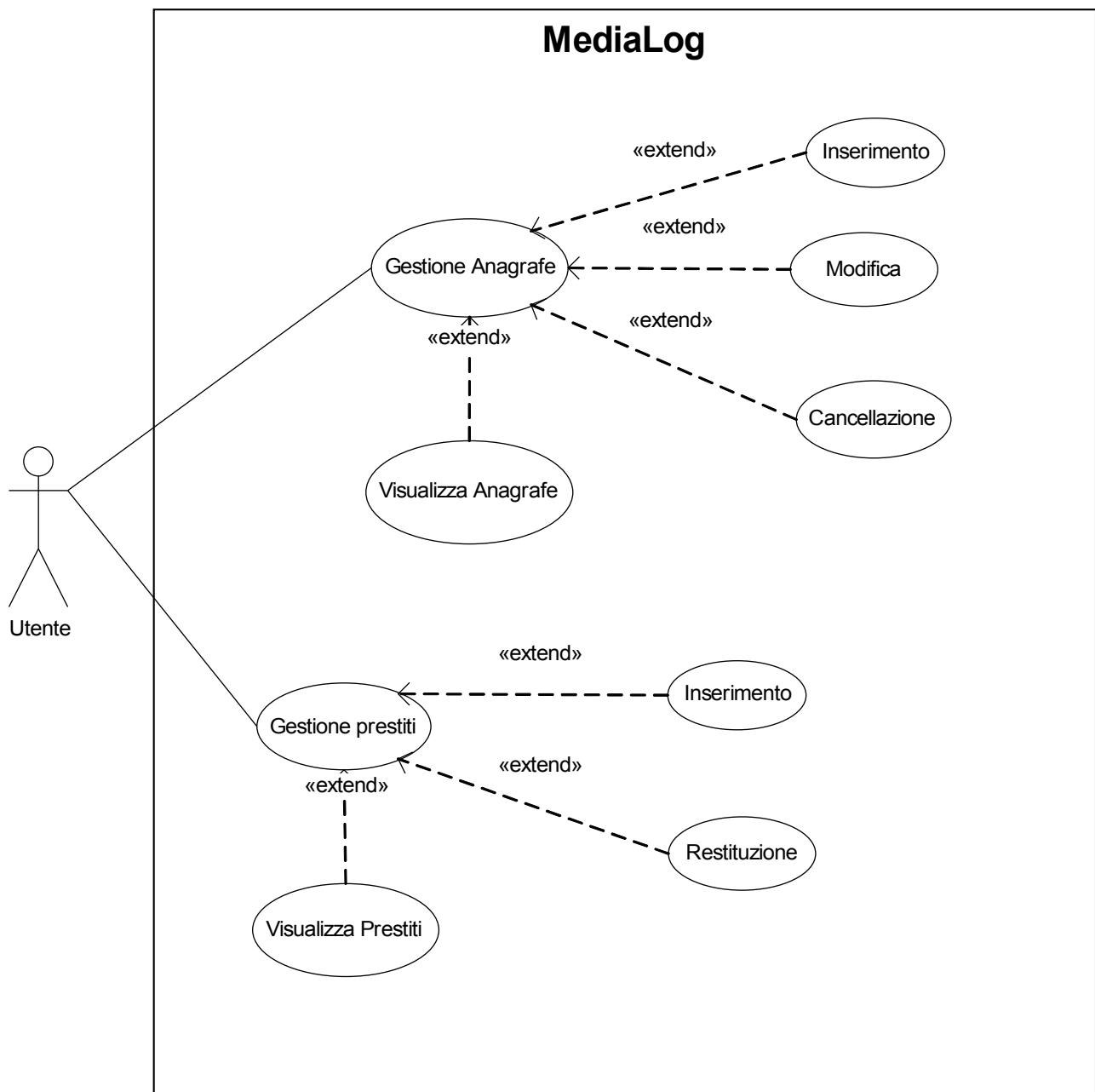


fig. 2: casi d'uso Prestiti / Anagrafe

Scenari

Scenari dei casi d'uso più significativi:

- Aggiunta supporto al catalogo
- Ricerca
- Impostazione chiave di ricerca
- Inserimento di un nuovo prestito

SCENARIO 1.0 - Aggiunta supporto

Attori:

- Utente

Pre-condizioni:

- E' inserito un supporto all'interno del computer.

Flusso principale:

- 1) L'utente seleziona il percorso su cui effettuare la scansione.
- 2) L'utente comunica il tipo di supporto.
- 3) Il sistema calcola l'identificativo da assegnare al supporto.
- 4) Il sistema effettua la scansione del contenuto e lo memorizza.
- 5) Il sistema comunica l'identificativo all'utente

Flusso alternativo:

- 1.a) Percorso non valido -> Il sistema ritorna al punto 1.
- 4.a) La scansione fallisce -> Il sistema avvisa l'utente. Si esce dallo scenario.

SCENARIO 2.0 - Ricerca:

Attori:

- Utente

Pre-condizioni:

- Il Catalogo non deve essere vuoto.

Flusso principale:

- 1) Il sistema richiede all'utente di impostare una o più chiavi di ricerca [-> 3.0].
- 2) L'utente avvia la ricerca.
- 3) Il sistema effettua la ricerca e produce il risultato.
- 4) Il sistema visualizza il risultato.

Flusso alternativo:

- 3.a) Nel caso la ricerca abbia risultato vuoto, il sistema lo comunica e ritorna al punto 1.

SCENARIO 3.0 - Impostazione chiave di ricerca:

Attori:

- Utente

Pre-condizioni:

-

Flusso principale:

- 1) L'utente seleziona le chiavi di ricerca di interesse.
- 2) L'utente immette i valori per i campi selezionati al punto 2.

Flusso alternativo:

- 2.a) L'utente lascia il campo vuoto nonostante lo abbia selezionato -> il sistema genera un messaggio d'errore adeguato e ritorna al punto 2.

SCENARIO 4.0 - Inserimento di un nuovo prestito:

Attori:

- Utente

Pre-condizioni:

- Esiste un supporto in archivio
- Esiste una persona registrata nell'anagrafe

Flusso principale:

- 1) L'utente fornisce l'identificativo del supporto.
- 2) L'utente fornisce il nickname della persona ricevente il supporto.
- 3) Il sistema calcola la data odierna
- 4) Il sistema memorizza il prestito.

Flusso alternativo:

- 1.a) Il nome del supporto non è valido -> Il sistema richiede l'inserimento e torna al punto 1.
- 1.b) Il supporto è già in prestito -> Il sistema lo comunica e si esce dallo scenario.
- 2.a) L'utente fornisce un nickname non esistente nell'anagrafe -> Il sistema richiede l'inserimento del nome e torna al punto 2.

Modello statico

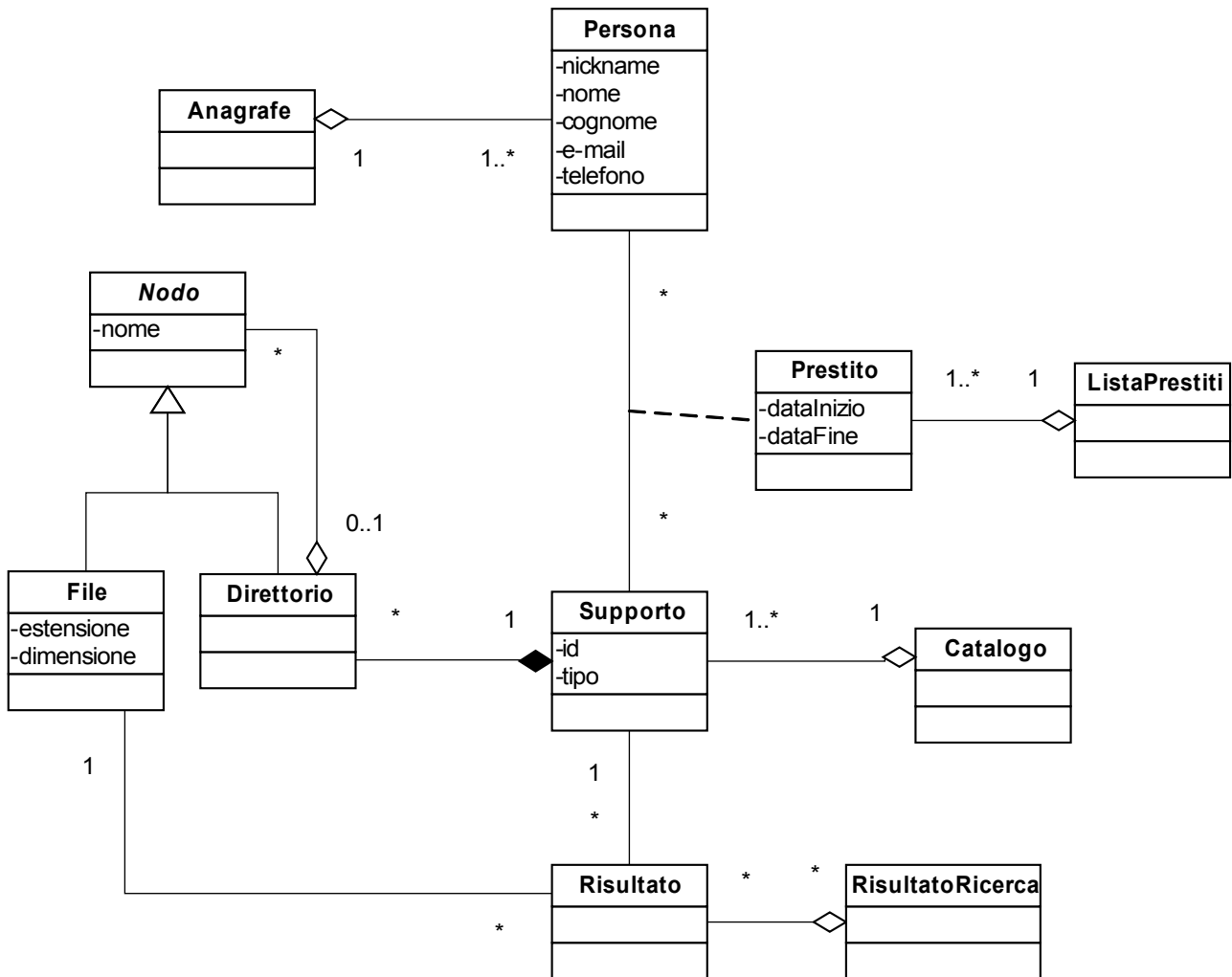


fig. 3: modello statico

Dal documento dei requisiti sono state ricavate le classi del modello statico.

La classe Supporto contiene le informazioni relative al supporto da catalogare. La classe Catalogo è una aggregazione di Supporto poiché rappresenta un contenimento logico, mentre la classe Supporto è una composizione di Direttorio poiché rappresenta un contenimento fisico: l'eliminazione di un supporto comporta anche l'eliminazione di tutto l'albero del filesystem.

Il filesystem è rappresentato da una struttura dati ad albero realizzata dalla classe astratta Nodo da cui derivano le classi File e Direttorio.

Un Prestito è rappresentato da una classe associazione che lega un Supporto a una Persona.

Il Risultato di una ricerca è modellato da una classe che contiene un'associazione ad un Supporto e ad un File.

Modello Dinamico

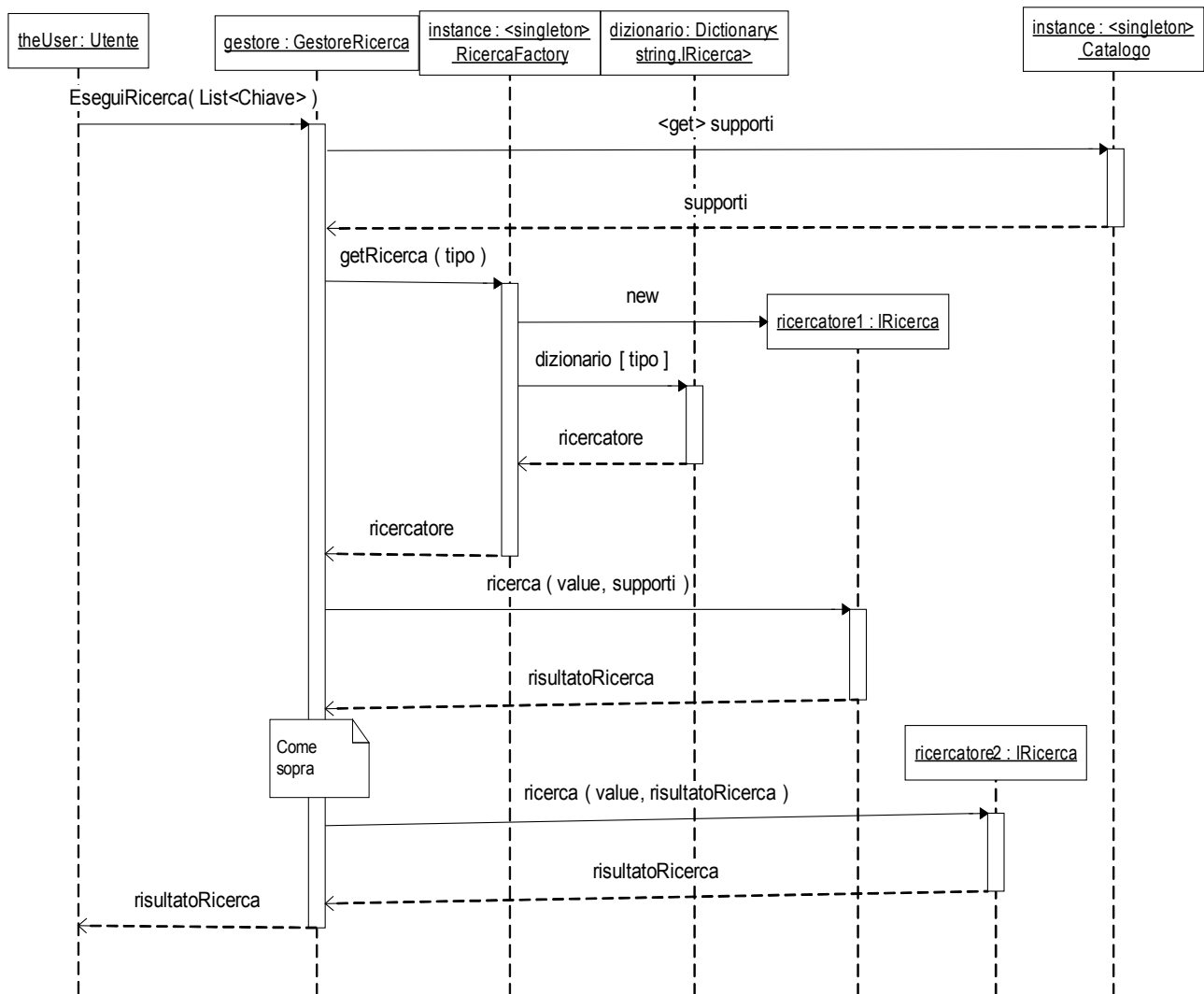


fig. 4: diagramma di sequenza della ricerca

Questo diagramma mostra la sequenza di messaggi scambiati tra gli oggetti durante l'esecuzione di una ricerca.

L'utente chiama il metodo statico EseguiRicerca() del GestoreRicerca passando la lista di Chiavi di ricerca. La classe GestoreRicerca ha la responsabilità di coordinare tutte le operazioni che portano al risultatoRicerca.

In primo luogo ottiene dal Catalogo la lista dei supporto, poi richiede a RicercaFactory un'istanza di IRicerca del tipo specificato nella chiave. L'istanza di IRicerca ottenuta implementa un particolare algoritmo di ricerca e ha la responsabilità di effettuarla restituendo un risultatoRicerca al GestoreRicerca.

In caso di chiavi multiple il GestoreRicerca ottiene una nuova istanza di IRicerca che effettua la ricerca sul risultatoRicerca parziale precedente.

Progettazione

Note

A partire dal modello statico abbiamo sviluppato i diagrammi di progettazione introducendo le seguenti modifiche:

- Implementazione delle associazioni, tramite l'introduzione di eventuali classi contenitore. Per queste ultime si è deciso di utilizzare la classe generica List<>
- Determinazione delle navigabilità delle associazioni
- D'altro canto nel caso di ricerca della persona a cui è stato prestato il supporto è necessario scorrere l'intera lista di Prestiti ma si suppone che le dimensioni siano piccole e i tempi di ricerca siano bassi.
- Introduzione di nuove classi contenenti la logica applicativa, come ad esempio Scanner e Identificatore
- Definizione dei tipi di dato dei campi delle classi utilizzando il linguaggio di programmazione C#
- Introduzione delle proprietà per accedere ai campi privati degli oggetti
- Applicazione di principi di progettazione e di design pattern (vedi capitolo “Pattern”)
- Definizione di un entry point dal quale è possibile accedere a tutte le classi del sistema

Diagrammi UML

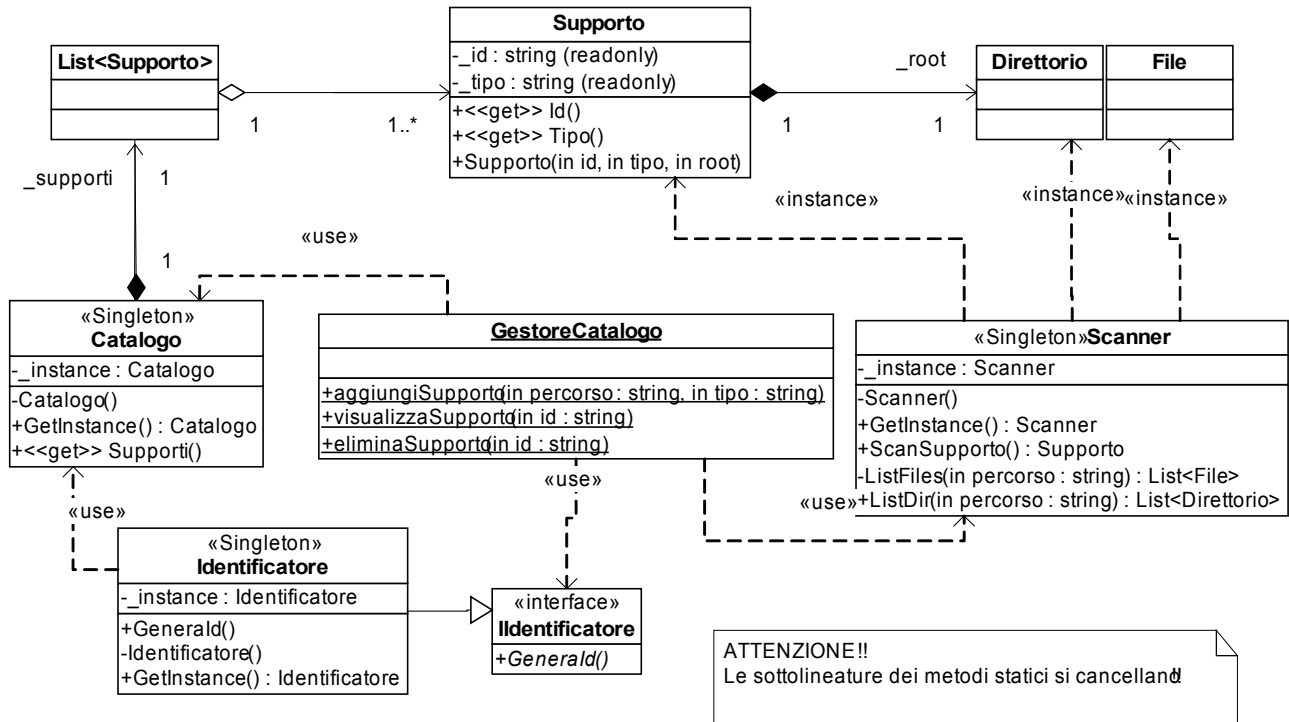


fig. 5: diagramma delle classi relativo alla gestione del catalogo

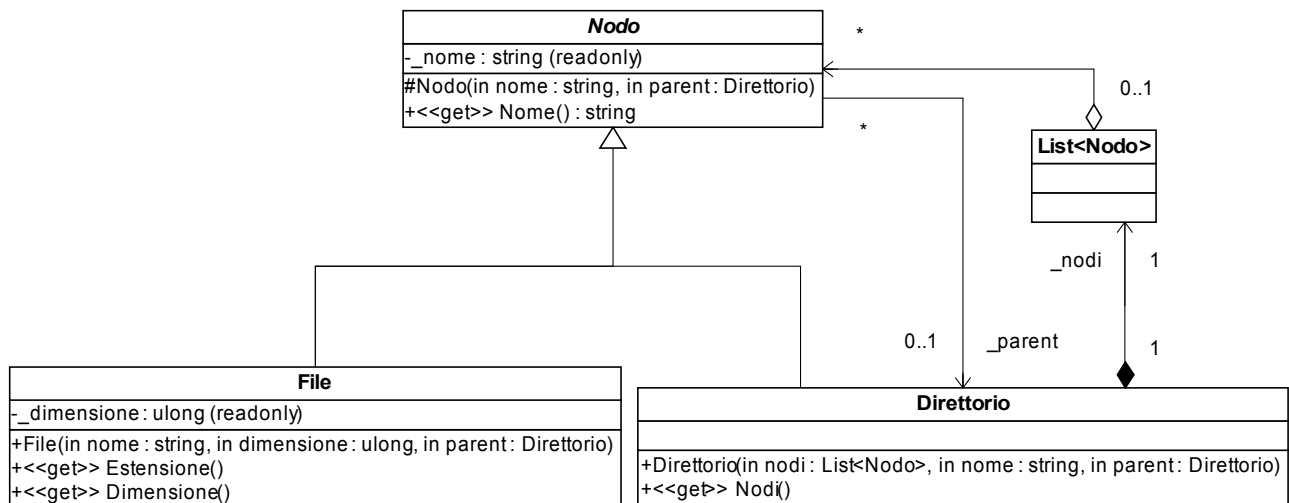


fig. 6: diagramma delle classi relativo al filesystem di un supporto

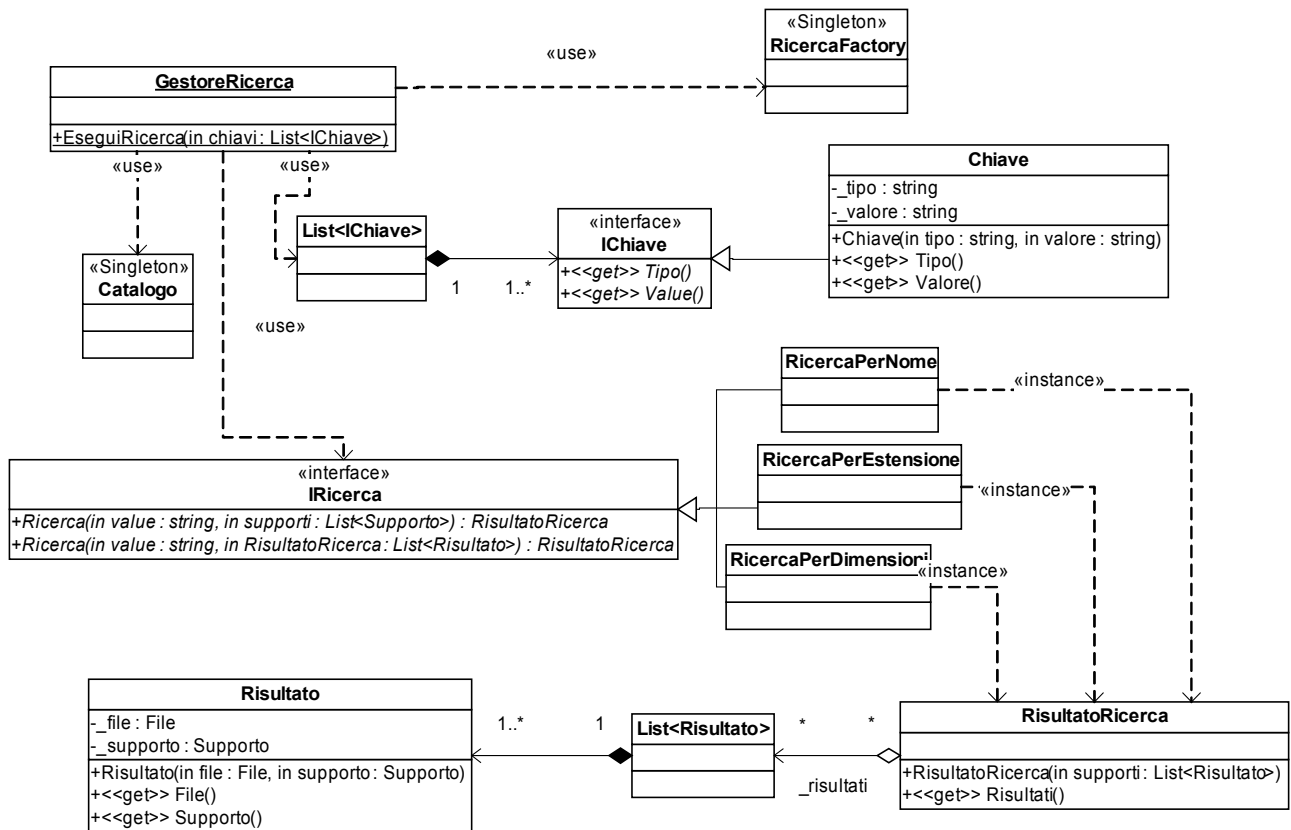


fig. 7: diagramma delle classi relativo alla ricerca di un file

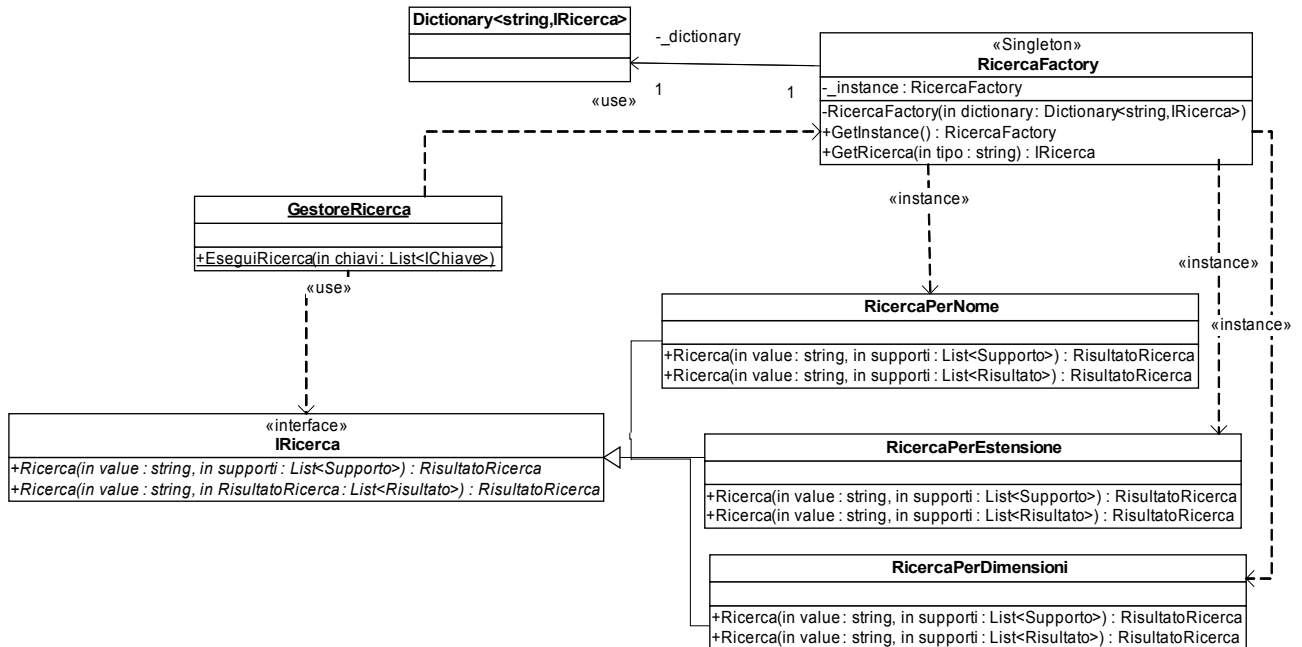


fig. 8: diagramma delle classi relativo alla gestione degli algoritmi di ricerca

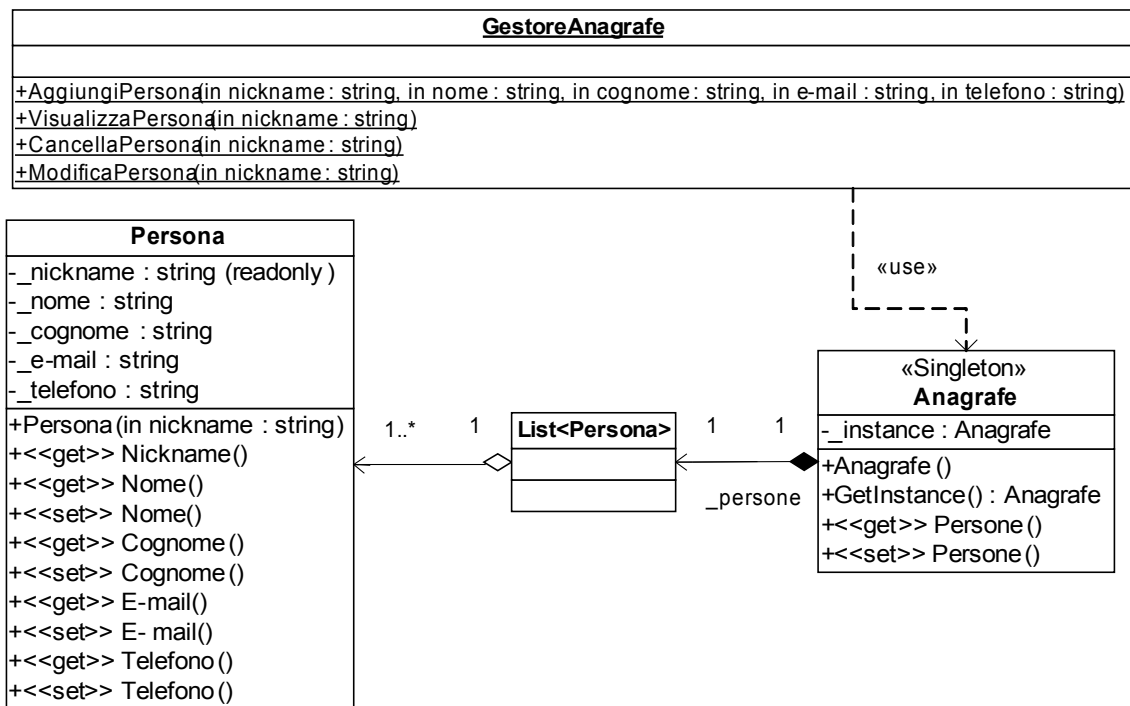


fig. 9: diagramma delle classi relativo alla gestione dell'anagrafe delle persone

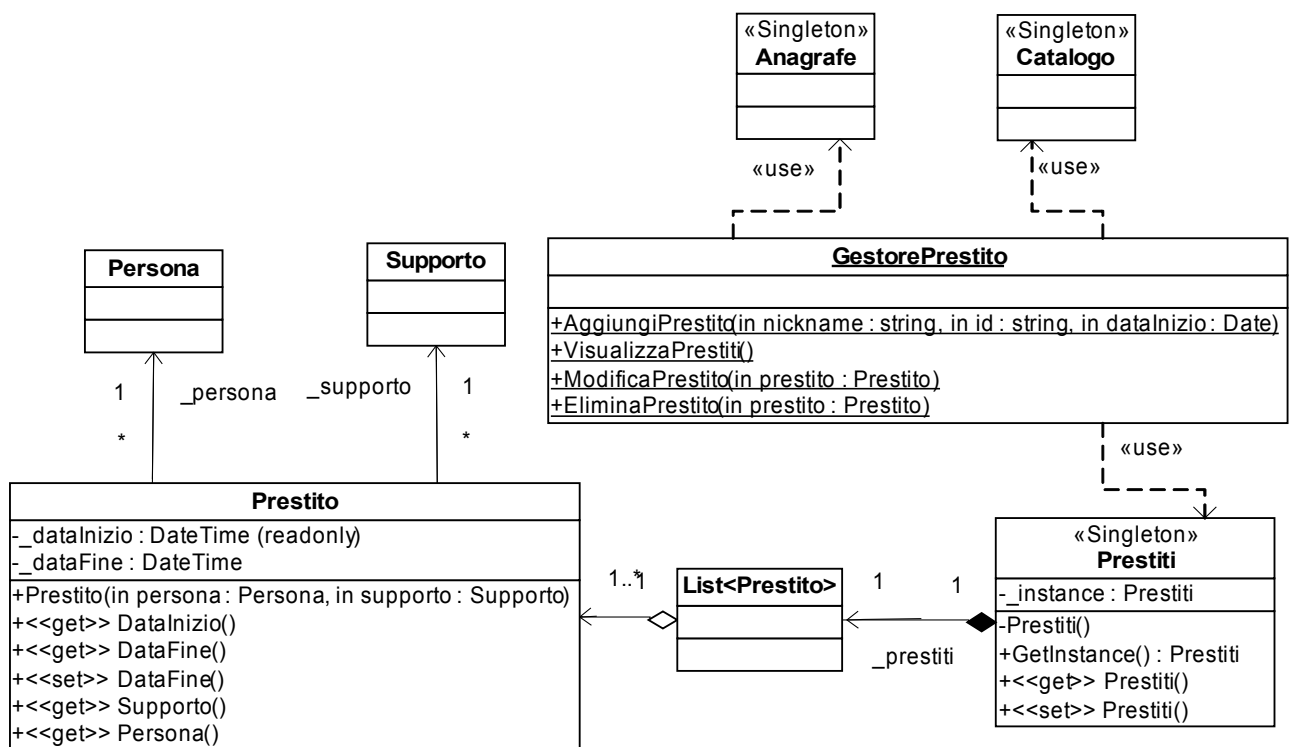


fig. 10: diagramma delle classi relativo alla gestione dei prestiti.

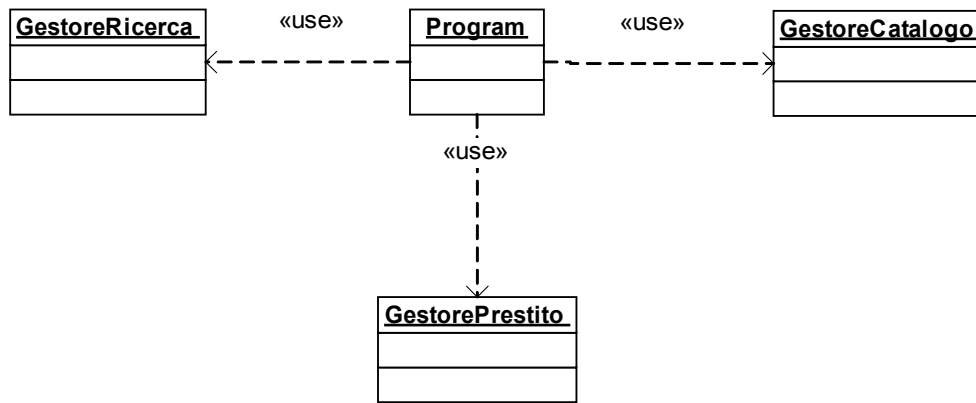


fig. 11: diagramma delle classi relativo all'entry point del sistema

Pattern

Composite

"Ogni supporto ha un filesystem strutturato in direttori e file: un direttorio può contenere altri direttori e/o file."

Il contenuto dei ogni supporto è quindi strutturato come un albero n-ario che permette di rappresentare il filesystem come una gerarchia di oggetti contenitori/contenuti.

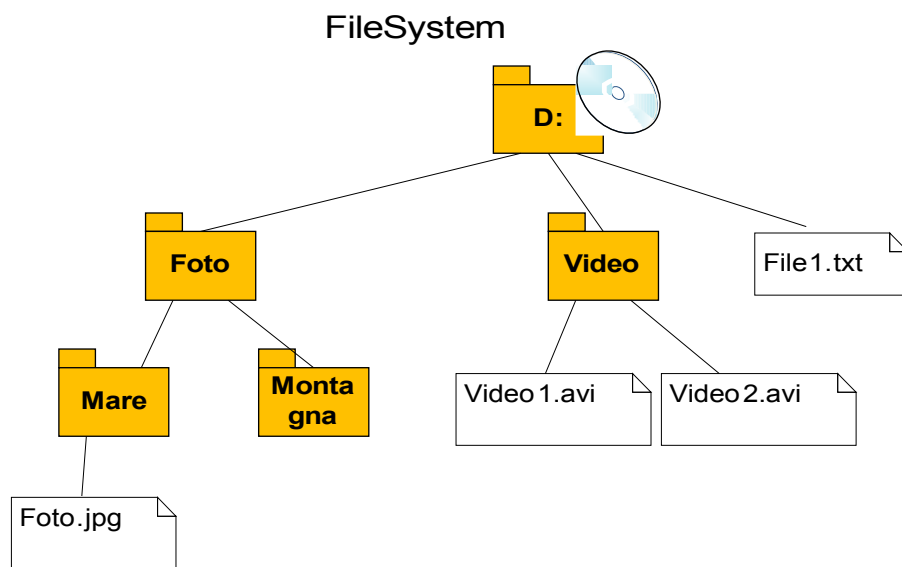


fig. 12: albero di un filesystem

Si è quindi deciso di utilizzare il pattern COMPOSITE in modo tale da trattare in modo uniforme oggetti singoli e oggetti composti.

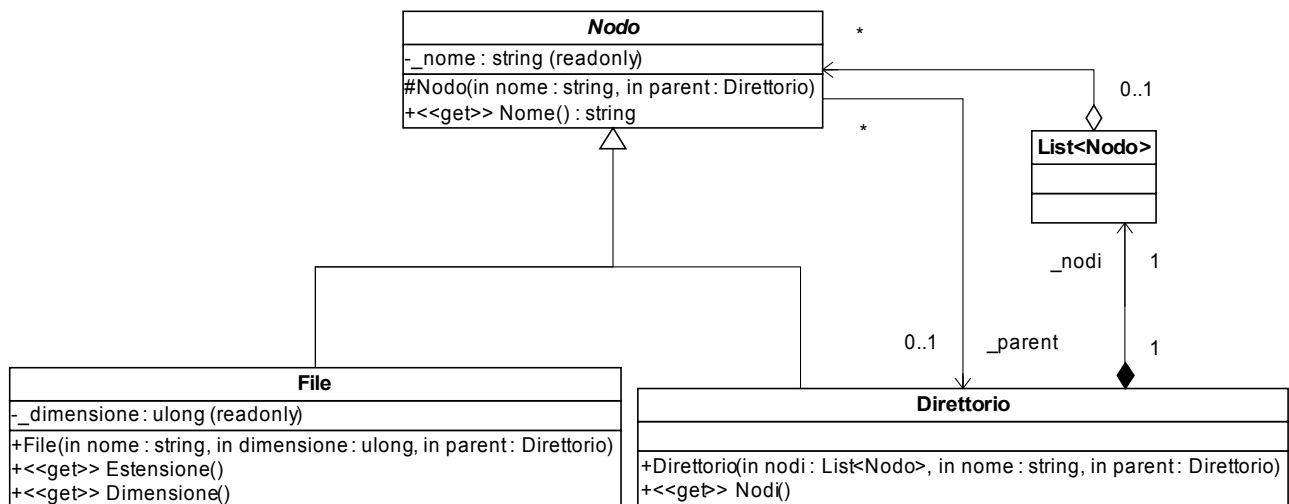


fig. 13: applicazione del pattern composite

L'elemento Component è il `Nodo`, la classe astratta che contiene i campi presenti in tutti gli elementi della gerarchia (nel nostro caso il "nome").

L'elemento Leaf è il `File` che infatti non ha figli nella gerarchia e contiene i campi che lo descrivono ("estensione" e "dimensione")

L'elemento Composite è il `Direttorio` che descrive gli elementi che possono avere figli. I figli possono essere altri `Direttorio` o `File`.

Il Client è il Supporto che contiene una lista di `Nodo`.

La realizzazione della associazioni uno a molti avviene tramite una classe contenitore. Inoltre in `Nodo` è presente un riferimento opzionale al `Direttorio` che lo contiene per velocizzare la navigazione all'interno del filesystem; tale riferimento deve essere coerente con la lista di `Nodi` contenuti nel `Direttorio`.

Strategy

"La ricerca può essere effettuata in base al nome del file, all'estensione oppure alla dimensione."

L'algoritmo di ricerca non è quindi fisso, ma cambia in base al vincolo inserito dall'utente.

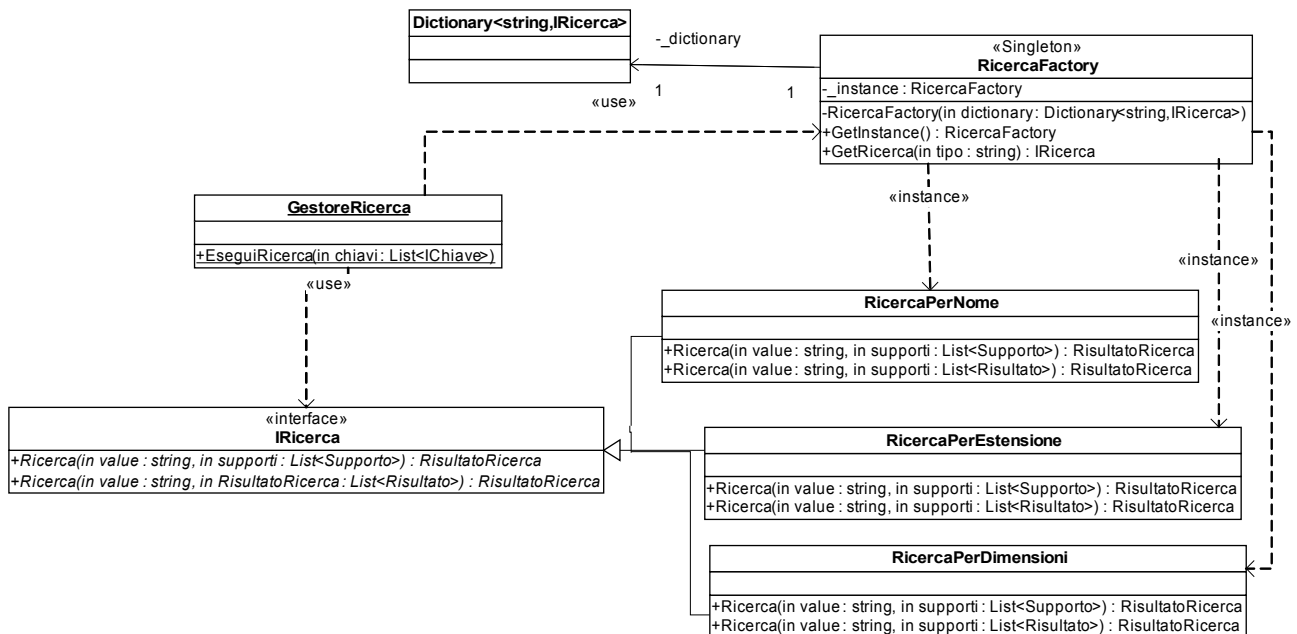


fig. 14: applicazione del pattern strategy

Si utilizza quindi un pattern Strategy che consente di definire un insieme di algoritmi tra loro correlati rendendoli intercambiabili. In questo modo sarà possibile inserire facilmente anche nuovi algoritmi di ricerca in futuro.

Factory

La classe **RicercaFactory** si occupa di creare l'istanza corretta dell'oggetto che implementa **IRicerca**. La factory conosce le classi che deve creare e reperisce da un dizionario le informazioni di associazione tra nome e classe concreta da istanziare: in questo modo si evita di cambiare il metodo `getRicerca()` quando si vogliono aggiungere ulteriori criteri di ricerca.

Principi di progettazione

Dependency Inversion Principle (DIP)

Il DIP consente, attraverso l'uso di astrazioni, di garantire il disaccoppiamento tra moduli di alto e basso livello.

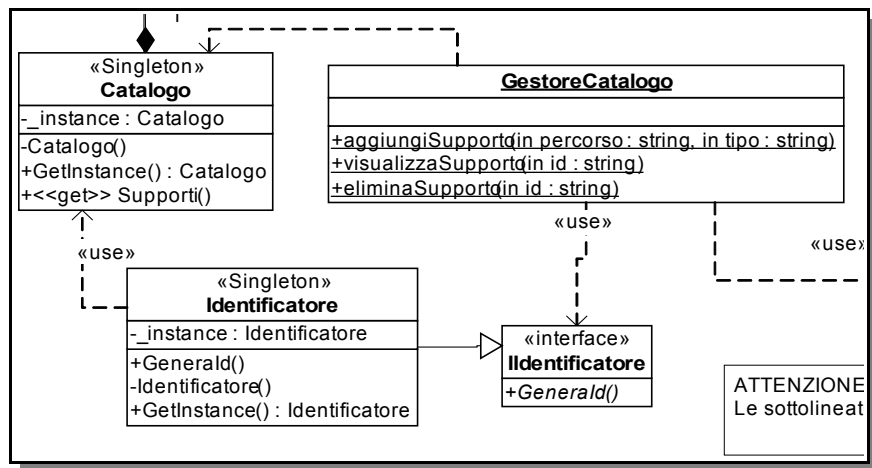


fig. 15: particolare riguardante l'uso del DIP

Ad esempio, per generare gli ID, il **GestoreCatalogo** utilizza un **Identificatore**. Nel caso in cui si avesse un riferimento diretto all'**Identificatore** e si volesse modificare quest'ultimo sarebbe necessario modificare anche la logica di **GestoreCatalogo**, violando l'Open Closed Principle.

Si è quindi deciso di introdurre una barriera di astrazione (**IIdentificatore**) che separa i due moduli.

Lo stesso è stato fatto con **Chiave** e **IChiave**.