



Specifica tecnica

Versione 1.0

Stato del documento:

*Formale ed
Esterno*

Sommario :

Questo documento si prefigge di illustrare l'architettura del prodotto.

Redazione

Nominativo	Ruolo	Data
Eric Miotto	Progettista	28 gennaio 2007
Stefano Gazzola	Progettista	28 gennaio 2007

Lista di Distribuzione

Nominativo	Ruolo
Tullio Vardanega	Committente
Renato Conte	Committente
Eric Miotto	Progettista
Margherita Collicelli	Progettista
Lucia Meneghello	Amministratore
Roberto Pordon	Responsabile
Stefano Gazzola	Progettista
Alberto Meneghello	Verificatore
Lorenzo Daniele	Verificatore

Approvato da:

Versione	Nominativo	Data
1.0	Roberto Pordon	02 febbraio 2007

Registro delle Modifiche:

Versione	Autore	Data
1.0	Roberto Pordon	02 febbraio 2007
Approvazione documento per la Revisione Preliminare di Progetto		
0.21	Alberto Meneghello, Lorenzo Daniele	01 febbraio 2007
Verifica del documento. Corrette alcune discrepanze sintattiche.		
0.20	Stefano Gazzola	01 febbraio 2007
Integrazione capitolo 3.1.4.		
0.19	Stefano Gazzola	01 febbraio 2007
Controllo e correzione generale.		
0.18	Stefano Gazzola	31 gennaio 2007
Inserita matrice di tracciamento componenti-requisiti, integrazione del formalismo di specifica con la spiegazione della nomenclatura utilizzata. Inserimento descrizione classi e diagrammi delle classi.		
0.17	Eric Miotto	31 gennaio 2007
Inserita prima bozza su WS-Didattica (sezione 3.2). Diviso in due sezioni la descrizione del metodo e del formalismo di specifica (sezioni 2.1 e 2.2).		
0.16	Stefano Gazzola	30 gennaio 2007
Modifiche capitolo 3.2		
0.15	Lucia Meneghello	30 gennaio 2007
Operate correzioni ortografiche e formali al testo. Aggiornamento Lista di distribuzione.		
0.14	Alberto Meneghello, Lorenzo Daniele	29 gennaio 2007
Aggiunti termini al glossario.		
0.13	Eric Miotto	28 gennaio 2007
Operata ulteriore correzione del testo. Abbozzata le descrizioni dei componenti nella sezione . Aggiunta sezione sulla collaborazione (sezione 1.3)		
0.12	Eric Miotto	28 gennaio 2007
Prima correzione del testo.		
0.11	Stefano Gazzola	28 gennaio 2007
Inserito capitolo 1 (1.1, 1.2) e capitolo 2 (2.1, 2.2)		
0.1	Eric Miotto	29 novembre 2006
Prima bozza del documento		

Indice

1	Introduzione.....	5
1.1	Scopo del documento.....	5
1.2	Scopo del prodotto.....	5
1.3	Collaborazione con Swell Systems.....	5
1.4	Riferimenti.....	5
2	Definizione del prodotto.....	5
2.1	Metodo di specifica.....	5
2.2	Formalismo di specifica.....	6
2.3	Presentazione dell'architettura generale del sistema e identificazione dei componenti architettonici di alto livello.....	6
2.3.1	<i>Studio di fattibilità tecnologica a supporto dell'architettura scelta.....</i>	<i>8</i>
3	Descrizione dei singoli componenti.....	10
3.1	APP Didattica [CSA-01].....	10
3.1.1	<i>Tipo, obiettivo e funzione.....</i>	<i>10</i>
3.1.2	<i>Relazioni d'uso di altre componenti.....</i>	<i>12</i>
3.1.3	<i>Interfacce con e relazioni di uso da altre componenti.....</i>	<i>13</i>
3.1.4	<i>Attività svolte e dati trattati.....</i>	<i>13</i>
3.2	WS-Didattica [CSW-01].....	13
3.2.1	<i>Tipo, obiettivo e funzione.....</i>	<i>13</i>
3.2.2	<i>Relazioni d'uso di altre componenti.....</i>	<i>13</i>
3.2.3	<i>Interfaccia esposta verso altre componenti.....</i>	<i>13</i>
4	Stime di fattibilità e di bisogno di risorse.....	25
5	Tracciamento della relazione componenti - requisiti.....	25

1 Introduzione

1.1 Scopo del documento

Questo documento si propone di definire l'architettura del nostro sistema software ad un livello di astrazione molto alto al fine di capirne il funzionamento logico. Tale architettura sarà il mezzo attraverso cui cercheremo di raggiungere gli obiettivi di progetto.

Inoltre saranno definiti i componenti costitutivi e la loro organizzazione, individuando le interfacce che permettono ai componenti stessi di interagire fra loro.

1.2 Scopo del prodotto

Per lo scopo del prodotto fare riferimento a [AR].

1.3 Collaborazione con Swell Systems

L'intero prodotto viene sviluppato in collaborazione con l'altro gruppo che ha scelto il capitolato C04, Swell Systems. La collaborazione è dovuta a due motivi:

- il capitolato è tale per cui è auspicabile (e sensata) la produzione di un solo prodotto;
- le risorse e il tempo a disposizione di entrambe i gruppi non permettono di giungere ad un prodotto che soddisfi i requisiti.

Con la collaborazione si spera di costruire una buona infrastruttura che possa essere evoluta in futuro per arrivare alla costruzione di un ottimo sistema informativo che risponda ai requisiti del capitolato.

1.4 Riferimenti

- [AR] Analisi dei Requisiti, Egoless Group
- [G] Glossario, Egoless Group
- [PP] Piano di progetto, Egoless Group
- [UML] Specifica di UML, versione 1.4, OMG
<http://www.uml.org/>

2 Definizione del prodotto

2.1 Metodo di specifica

La specifica si svilupperà per gradi di astrazione partendo da un livello più alto in cui si illustra l'architettura suddividendola nelle sue componenti più significative che poi verranno via via smembrate per essere analizzate più dettagliatamente.

Questa specifica sarà integrata con nuove informazioni ogni qualvolta una nuova componente verrà aggiunta, analizzata o approfondita. In particolare in questa prima iterazione verrà preso in considerazione il modulo didattica (CU09). Questa scelta andrà ad influire anche sul lavoro di Swell Systems che darà maggiore priorità alle esigenze di questo modulo.

2.2 Formalismo di specifica

L'architettura del prodotto viene descritta nel seguente documento utilizzando diagrammi delle classi e diagrammi dei componenti espressi in *UML* 1.4 ([UML]) e costruiti con ArgoUML 0.24-Alpha 3 come strumento principale.

Per i diagrammi delle classi si adottano le seguenti convenzioni (oltre a quelle tipiche di UML):

- per ogni classe presentata verranno modellati solo gli attributi. Verrà sottinteso che questi attributi sono privati e che per ognuno di essi ci sarà una coppia di metodi get/set per accedere e modificare il valore;
- dato che ArgoUML permette di specificare la visibilità di un attributo ma non visualizza il simbolo appropriato nel diagramma e poiché nel codice è utile poter riconoscere se un attributo è privato dal nome, gli attributi privati inizieranno con un underscore.

Nella descrizione dei Web Service si modelleranno le classi che generano il Web Service e non il WSDL (Web Service Description Language).

Ciascun componente sviluppato da Egoless Group sarà identificato da una sigla con la seguente struttura:

CSx-y/z

dove "CS" sta per "Componente software", la variabile x può assumere il valore W (Web service) oppure A (Applicativo), la variabile y è un numero progressivo espresso con due cifre (01,02,...11,...), la variabile z indica la numerazione delle classi di ciascun componente con un numero progressivo(1,2,3,...).

2.3 Presentazione dell'architettura generale del sistema e identificazione dei componenti architetturali di alto livello

Data l'impossibilità di realizzare l'intero prodotto richiesto nel tempo pianificato (vedi [PP]), la scelta dell'architettura è stata fatta in modo tale da facilitare la manutenzione e il riuso a lungo termine, in modo che l'investimento fatto possa durare nel tempo.

L'architettura ad alto livello che sarà presentata nel seguito si basa su di un particolare *pattern architetturale* detto "multi-layer pattern" appartenente alla famiglia più generale del "layers pattern". La comunicazione tra i vari livelli avviene attraverso Web Service. L'uso dei Web Service permette di avere riuso e interoperabilità.

La scelta di questo tipo di pattern architetturale nasce dall'esigenza di decomporre e catalogare la struttura dell'applicazione in gruppi di *moduli* che per loro intrinseca natura hanno esigenze e bisogni comuni.

Individuati i moduli si cercherà, quindi, di isolarli e di renderli il più indipendenti possibile attraverso le tecnologie utilizzate.

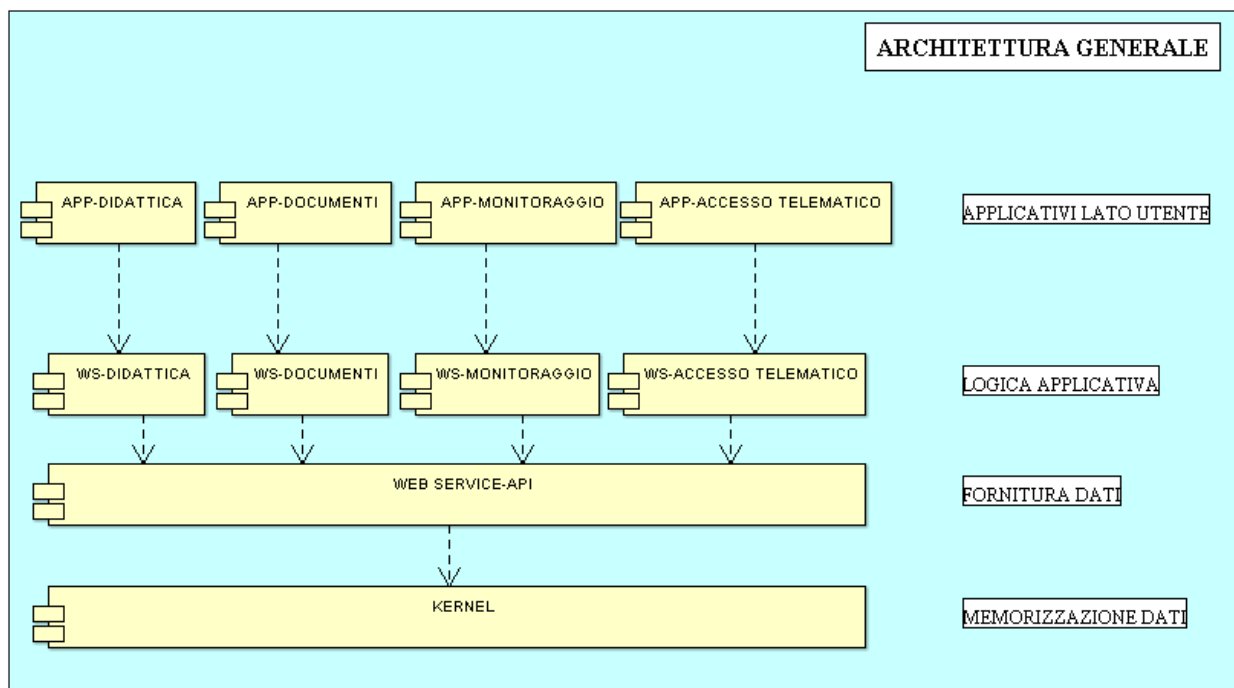


Illustrazione 1: Architettura generale del sistema

Qui di seguito è riportato il tracciamento dei componenti e il loro identificativo:

Componente	Sigla
APP-DIDATTICA	CSA-01
APP-DOCUMENTI	CSA-02
APP-MONITORAGGIO	CSA-03
APP-ACCESSO TELEMATICO	CSA-04
WS-DIDATTA	CSW-01
WS-DOCUMENTI	CSW-02
WS-MONITORAGGIO	CSW-03
WS-ACCESSO TELEMATICO	CSW-04

L'architettura individuata è schematizzata in Illustrazione 1 ed è formata dalle seguenti

componenti:

- APP-Didattica, APP-Documenti, APP-Monitoraggio, APP-Accesso telematico: insieme di applicativi lato *client* e *desktop-oriented*. Permettono all'utente di interagire con il sistema e di ottenere il servizio voluto. Ogni applicativo può essere installato su più macchine differenti. Si è scelto di sviluppare questi moduli come applicazioni desktop per poter realizzare un'interfaccia grafica *user-friendly* e possibilmente molto sofisticata.
- WS-Didattica, WS-Documenti, WS-Monitoraggio, WS-Accesso telematico: insieme di web services con i quali ciascun tipo di applicativo interagisce. L'utilizzo di web services permette di disaccoppiare l'interfaccia utente dalla parte logica permettendo un buon grado di isolamento e di riuso dei servizi. Questo permette di poter modificare, evolvere, sostituire l'implementazione del servizio in modo del tutto indipendente dai livelli superiori e inferiori.
- WEB SERVICE-API: interfaccia che sancisce quali informazioni i web services di livello superiore possono richiedere e sulle quali possono lavorare, indipendentemente da come tali informazioni verranno concretizzate come dati di un database. Questo livello si propone esso stesso a livello superiore come web service per rendere il servizio offerto facilmente adattabile a cambi strutturali e tecnologici (es. permettere un consorzio di scuole per la gestione in comune di certi servizi rivolti ai dati).
- *KERNEL*: insieme formato dal database che memorizza i dati della scuola e dalle funzionalità atte alla sua gestione.

2.3.1 Studio di fattibilità tecnologica a supporto dell'architettura scelta

L'architettura presentata precedentemente potrebbe risultare abbastanza utopistica nel caso in cui non venga supportata da uno studio di fattibilità tecnologica che possa giustificare le scelte fatte. Tali soluzioni e alcune di quelle scartate sono passate attraverso uno studio dell'impatto che potevano avere a livello fisico e prestazionale.

Le necessità dell'ambiente nel quale il nostro sistema andrà ad innestarsi hanno dettato questo tipo di comportamento

All'interno del sistema scuola infatti è stato individuato il requisito non funzionale RN14.B che impone di utilizzare configurazioni hardware spesso datate o comunque di poter prevedere l'innesto di configurazioni con un basso impatto economico sulle casse della scuola stessa

Allo stato attuale delle cose inoltre dobbiamo considerare che il nostro sistema potrebbe girare interamente all'interno di un'unica macchina: è inevitabile quindi che dovranno essere presenti nello stesso momento più sistemi software (*DBMS*, applicativi, *JVM*...) che rivendicheranno il proprio spazio e le proprie esigenze.

D'altro canto, è necessario porre dei requisiti hardware minimi tali da garantire il buon funzionamento del sistema.

Analizzando dall'alto l'architettura proposta incontriamo lo strato degli "applicativi utente" che in base alle nostre previsioni verranno sviluppate in linguaggio Java. Considerato che un IDE completo e molto complesso sviluppato in Java come NetBeans (www.netbeans.org) ha come requisiti di sistema un Pentium 3 con processore 500 *Mhz* e 512MB di *RAM* con valore di mercato piuttosto irrisorio, i nostri applicativi ben meno complessi necessiteranno di una configurazione hardware inferiore.

Un livello sotto incontriamo la "logica applicativa" basata su web-service che dovrà gestire accessi di tipo concorrenziale. Partendo dal presupposto che un limitato numero di persone avranno il diritto di accedere alle funzionalità proposte, ogni web-service (e quindi per ogni applicativo a livello superiore) dovrà gestire un numero di accessi contemporanei che nel caso peggiore risulta essere un numero molto basso (dell'ordine delle decine). Nel caso medio tale

numero sarà ancora più irrisorio. Con questo possiamo giustificare anche il fatto che nonostante il web-service utilizzi per le comunicazioni *XML* (quindi operando onerose serializzazioni/deserializzazioni da/verso XML) l'impatto viene tranquillamente assorbito proprio perché le richieste saranno molto diluite nel tempo.

Discorso duale può essere fatto per i *layer* sottostanti con l'unica nota rivolta alla base di dati e al DBMS che verranno utilizzati i quali dovranno gestire una base di dati abbastanza ridotta e quindi l'impatto di un DBMS come MySQL viene tranquillamente assorbito.

L'unica risorsa che deve essere massimizzata (sempre rispettando i limiti imposti) è la memoria Ram: tra i 512 MB e 1024 MB i costi allo stato attuale sono molto contenuti (di molto inferiori ai 100 €) quindi un piccolo investimento su questo fronte può essere assorbito tranquillamente nel tempo.

Per quanto riguarda il disco rigido anche qui capienze più che sufficienti (≤ 100 GB) si possono trovare con costi molto inferiori a 100 € anche se sono risorse che una scuola facilmente può recuperare in loco.

3 Descrizione dei singoli componenti

Verranno descritte le componenti legate ai requisiti obbligatori in [AR]. Non verranno descritti i moduli di competenza di Swell Systems.

3.1 APP Didattica [CSA-01]

3.1.1 Tipo, obiettivo e funzione

APP Didattica è l'applicazione desktop usata per gestire i dati di studenti , docenti ed infrastrutture. L'obiettivo principale è quello di fornire un'interfaccia grafica il più completa e semplice possibile.

Di seguito si trovano alcuni *screenshot* che rappresentano la prima bozza di interfaccia grafica che ha lo scopo di organizzare le funzionalità per avere un primo contatto con i possibili problemi a livello visivo che potrebbero sorgere.

Al fine di una corretta visualizzazione li abbiamo ruotati.

Gestione della Didattica
File Edit Help

Gestione Studenti | Gestione Personale | Gestione Infrastrutture | Registro Del Docente

Ricerca Studente

☐ Ricerca per matricola

☐ Ricerca per cognome-nome

☐ Ricerca per classe-sezione

Risultati Ricerca

Nome	Cognome	Matricola	Data di nascita	Classe	Sezione

Ordina per
☐ Nome
☐ Cognome
☐ Sezione
☐ Classe

Modifica-Inserimento dati dello studente

Matricola

Nome

Cognome

Data di nascita
 ...
 ...

Carriera studente

Materia1	Mater...	...	MEDIA	CREDITI	DEBITI

Illustrazione 2: Screenshot della GUI "gestione studenti" dell'applicativo didattico

Questa GUI presenta la schermata "Gestione studenti" che sarà utilizzata principalmente dal personale della segreteria per il trattamento dei dati sensibili dello studente

Gestione della Didattica File Edit Help

Gestione Studenti | Gestione Personale | Gestione Infrastrutture | **Registro Del Docente**

Ricerca Studente

☐ Ricerca per matricola ☐ Ricerca per cognome-nome ☐ Ricerca per classe-sezione

Avvia ricerca

Risultati Ricerca

Nome	Cognome	Matricola	Data di nascita	Classe	Sezione

Ordina per
☐ Nome
☐ Cognome
☐ Sezione
☐ Classe

Carica dati studente

Voti Studente

Id_voto	Data	Descrizi...	Tipo	Valore

Data: Tipo: Valore: Descrizione:

Assenze studente

Id	Data inizio	Data fine	Giustific...	Note

Id: Genera: Data inizio: Data fine: Note: ☐ Giustificazione

Illustrazione 3: Screenshot della parte “registro docente” dell'applicativo didattica

Questa GUI presenta la schermata “Registro docente” che sarà utilizzata principalmente dal docente per la gestione dei voti e delle assenze dello studente.

3.1.2 Relazioni d'uso di altre componenti

Il componente usa WS Didattica per applicare le modifiche che l'utente ha specificato. L'insieme

delle funzionalità che tale applicativo utilizzerà sono quelle messe a disposizione dall'interfaccia del Web Service.

3.1.3 Interfacce con e relazioni di uso da altre componenti

Il componente non è usato da nessun'altra componente perché si interfacerà direttamente con l'utente.

3.1.4 Attività svolte e dati trattati

Il componente sarà strutturato internamente in più parti per la produzione dei dati da passare al Web Service sottostante.

Saranno individuate e implementate le parti seguenti:

- una parte di GUI (vedi screenshot) per l'inserimento e la visualizzazione dei dati;
- una parte logica per un controllo preliminare della validità dei dati inseriti e per l'elaborazione dei dati da visualizzare.
- una parte di preparazione dei dati per lo scambio con il Web Service sottostante;

3.2 WS-Didattica [CSW-01]

3.2.1 Tipo, obiettivo e funzione

WS Didattica implementa tutta la logica per gestire i dati di docenti e studenti. In particolare:

- permette di gestire gli studenti e i professori;
- per ogni studente permette di gestire i voti e le assenze;
- permette di gestire informazioni di supporto (tipo di voto, tipo di assenza, ...)

3.2.2 Relazioni d'uso di altre componenti

Per svolgere le sue funzioni, WS-Didattica si appoggia a WEB SERVICE-API:

- per recuperare e salvare in modo permanente tutte le informazioni della scuola;
- per avere le strutture dati fondamentali con le quali svolgere il proprio lavoro.

Allo stato attuale di questo documento le strutture dati che sono condivise tra i due livelli possono presentare alcune discrepanze. Questa situazione è causata dal fatto che la definizione delle classi nei due livelli sono ancora molto fluide. Le due parti stanno lavorando per convergere verso una soluzione comune

3.2.3 Interfaccia esposta verso altre componenti

Il componente espone un'interfaccia Web Service usata da APP-Didattica.



Illustrazione 4: Elenco dei metodi esposti dall'interfaccia del web service

Le classi esposte dal Web Service possono essere divise nelle seguenti categorie:

- classi dati ereditate da WEB SERVICE API. Queste si dividono ulteriormente in
 - **classi semplici:** corrispondono ad informazioni contenute che si mantengono relativamente stabili nel tempo. Per ogni classe ci sono poche istanze
 - **classi complesse:** corrispondono ad informazioni rilevanti per la scuola, che sono molto grandi e che mutano continuamente nel tempo. Per ogni classe ci sono molte istanze.

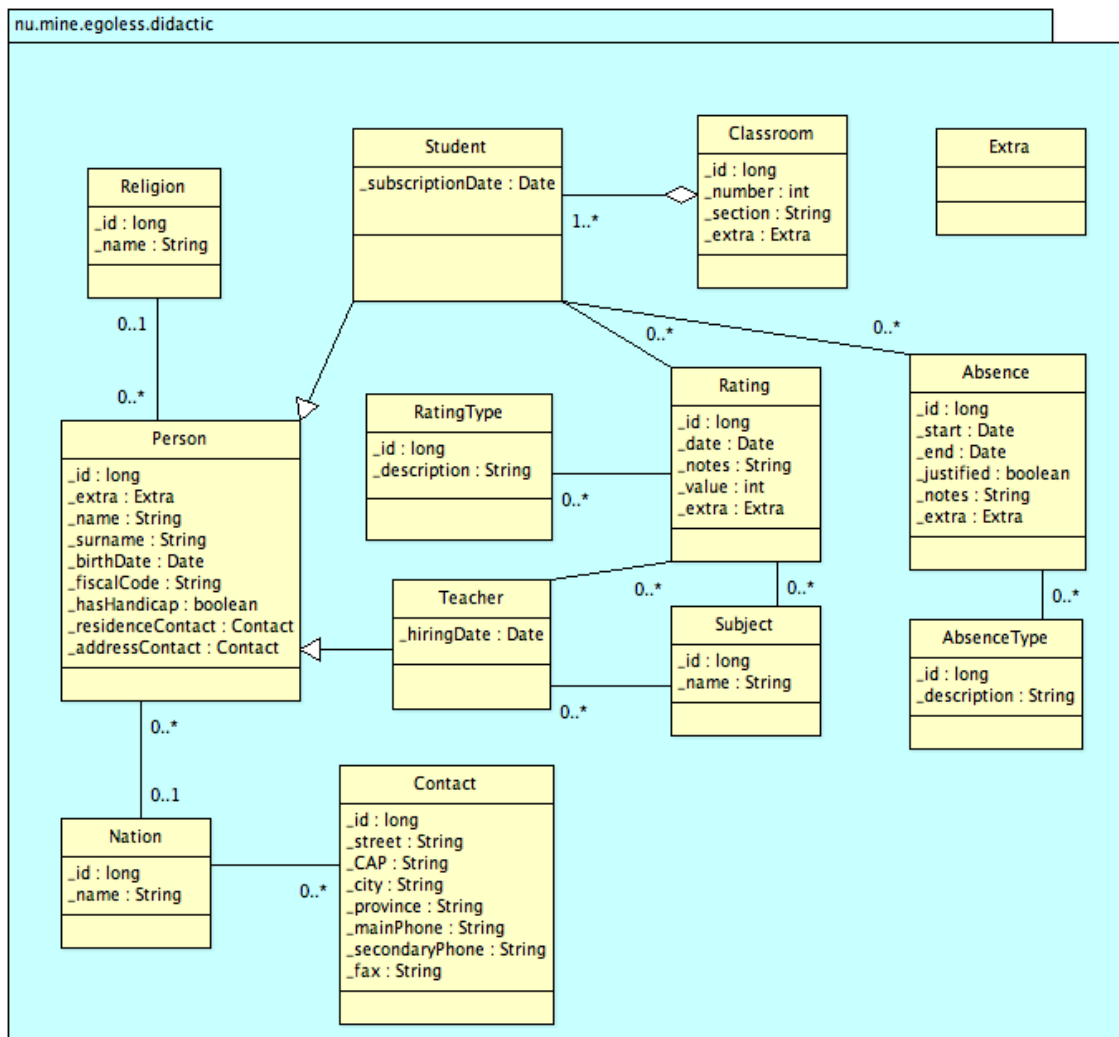


Illustrazione 5: Diagramma delle classi del WS-Didattica

- classi per la ricerca.** Sono strutture dati che contengono tutti i parametri per eseguire una data ricerca su oggetti di classi complesse. Sono previste per poter modificare agevolmente i criteri di una ricerca. Raccogliere tutti i criteri di una ricerca in una classe piuttosto che lasciarli come argomenti della funzione di ricerca permette di aggiungere o cambiare i criteri di ricerca senza alterare i parametri della funzione (agevolando così la manutenzione).

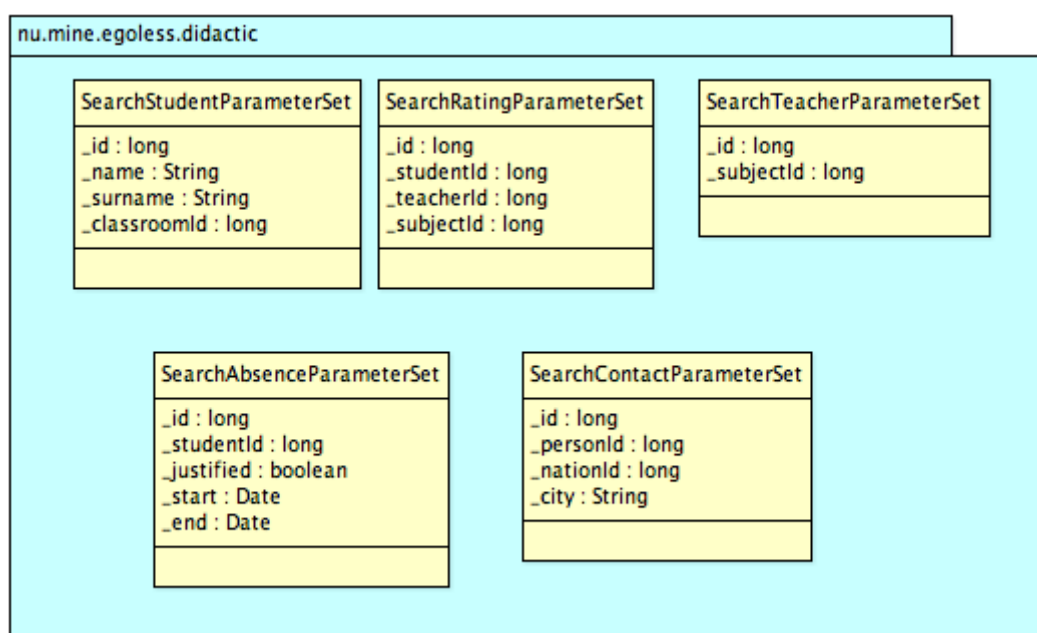


Illustrazione 6: Diagramma delle classi per la ricerca

Ogni classe maneggiata presenta un attributo `_id` che identifica univocamente tutti i suoi oggetti. Questo permette di avere un'equivalente del puntatore ad un oggetto a livello di Web Service e di rendere efficiente l'invio di messaggi (per esempio, per la cancellazione di un elemento basterà indicare il suo id e non ripassare inutilmente tutto l'elemento).

Per ogni classe complessa esposta vengono forniti i seguenti metodi nell'interfaccia del Web Service:

- aggiunta di un elemento;
- modifica di un elemento. Abbiamo preferito separare le operazioni di aggiunta e modifica per rendere più chiara l'interfaccia del Web Service;
- rimozione di un elemento esistente;
- ricerca di un insieme di elementi in base a criteri. In questo modo si cerca di minimizzare il traffico tra applicazione e il Web Service.

Per ogni classe semplice vengono invece forniti i seguenti metodi:

- aggiunta di un nuovo elemento;
- rimozione di un elemento esistente;
- recupero di tutti gli elementi.

Le ragioni di questa scelta sono:

- Dato che le classi semplici contengono pochi (due/tre attributi) e stabili nel tempo (es. Tipi di giustificazioni) non è necessario avere un metodo di modifica. Se un oggetto contiene dati da modificare è più conveniente cancellarlo e crearne uno nuovo con i dati corretti.
- Non serve un metodo di ricerca per le classi semplici. Considerato che gli oggetti di una classe semplice sono pochi, ogni oggetto contiene pochi attributi e stabili nel tempo, è sensato caricare tali dati all'avvio dell'applicazione.

Per le classi semplici per cui le istanze sono già decise è presente solo l'ultimo metodo.

Per ogni classe semplice esposta verranno invece forniti i metodi:

- aggiunta di un elemento;
- modifica di un elemento;
- rimozione di un elemento esistente;
- ricerca di un insieme di elementi in base a criteri.

Per le strutture semplici per cui le istanze sono già decise è presente solo l'ultimo metodo.

Segue ora un elenco di classi e dei metodi del Web Service. Per ogni classe si ometterà l'attributo id.

CSW-01/1 - Extra - classe	
<i>Breve descrizione:</i>	Questa classe rappresenta un insieme di informazioni aggiuntive generiche. Probabilmente sarà un insieme di oggetti o di stringhe. Viene impiegato nella classi complesse che potrebbero variare di struttura nel tempo a causa di requisiti cambiati. Extra non è ne' classe semplice ne' classe complessa.
<i>Attributi:</i>	-
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/3 - Nation - classe semplice	
<i>Breve descrizione:</i>	Rappresenta una nazione. Questa classe è importante per fissare i valori validi per i campi Nazione.
<i>Attributi:</i>	<ul style="list-style-type: none"> • name: nome della nazione
<i>Eredita da:</i>	-
<i>Legata a:</i>	Person, Contact

CSW-01/4 - Religion - classe semplice

<i>Breve descrizione:</i>	Rappresenta una religione. Questa classe è importante per fissare i valori validi per i campi Religione.
<i>Attributi:</i>	<ul style="list-style-type: none"> • name: nome della religione.
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Person

CSW-01/5 - Contact - classe semplice

<i>Breve descrizione:</i>	Racchiude le informazioni su come raggiungere una persona (principalmente indirizzo e telefono).
<i>Attributi:</i>	<ul style="list-style-type: none"> • street: via • CAP: Codice di Avviamento Postale • city: città • province: provincia • mainPhone: numero di telefono principale • secondaryPhone: altro numero di telefono • fax: numero di fax
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Person

CSW-01/6 - Student - classe complessa

<i>Breve descrizione:</i>	Rappresenta uno studente della scuola. subscriptionDate: data di iscrizione alla scuola
<i>Attributi:</i>	<ul style="list-style-type: none"> • subscriptionDate: data di iscrizione alla scuola
<i>Eredita da:</i>	<ul style="list-style-type: none"> • Person
<i>Legata a:</i>	<ul style="list-style-type: none"> • Classroom • Absence • Rating

CSW-01/7 - Classroom - classe semplice

<i>Breve descrizione:</i>	Rappresenta una classe come insieme di alunni. Il nome è un po' fuorviante (sembra che si parli dell'aula) ma la parola Class è già una parola chiave importante per i linguaggi di programmazione, per cui si è ritenuto opportuno non impiegarla.
<i>Attributi:</i>	<ul style="list-style-type: none"> • number: numero classe • section: sezione della classe • extra: informazioni extra
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/8 - Rating - classe complessa

<i>Breve descrizione:</i>	Rappresenta una valutazione assegnato ad uno studente in una materia.
<i>Attributi:</i>	<ul style="list-style-type: none"> • date: data in cui è stato registrato il voto • notes: campo in cui inserire annotazioni di varia natura • value: valore numerico del voto • extra: informazioni aggiuntive
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Studente • Rating • Type

CSW-01/9 - RatingType - classe semplice

<i>Breve descrizione:</i>	Specifica il tipo di valutazione. E' pensato per distinguere i normali voti dai voti di fine periodo (trimestre, quadrimestre) e fine anno.
<i>Attributi:</i>	<ul style="list-style-type: none"> • description: descrizione del tipo di valutazione
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Rating

CSW-01/10 - Subject - classe semplice

<i>Breve descrizione:</i>	Rappresenta una materia.
---------------------------	--------------------------

<i>Attributi:</i>	<ul style="list-style-type: none"> • name: nome della materia
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Teacher • Rating

CSW-01/11 - Teacher - classe complessa

<i>Breve descrizione:</i>	Rappresenta un professore della scuola.
<i>Attributi:</i>	<ul style="list-style-type: none"> • hiringDate: data di assunzione
<i>Eredita da:</i>	<ul style="list-style-type: none"> • Person
<i>Legata a:</i>	<ul style="list-style-type: none"> • Teacher • Rating

CSW-01/12 - Absence - classe complessa

<i>Breve descrizione:</i>	Rappresenta l'assenza di uno studente.
<i>Attributi:</i>	<ul style="list-style-type: none"> • start: giorno/ora in cui è iniziata l'assenza • end: giorno/ora in cui è finita l'assenza • justified: indica se l'assenza è stata giustificata o meno • notes: annotazioni aggiuntive • extra: informazioni aggiuntive
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • AbsenceType • Student

CSW-01/13 - AbsenceType - classe semplice

<i>Breve descrizione:</i>	Rappresenta una tipologia di assenza. Si è ritenuto opportuno introdurre questa classe per catalogare e rendere immediatamente fruibili le varie tipologie di assenza.
<i>Attributi:</i>	<ul style="list-style-type: none"> • description: descrizione della tipologia di assenza
<i>Eredita da:</i>	-
<i>Legata a:</i>	<ul style="list-style-type: none"> • Absence

CSW-01/14 - SearchStudentParameterSet - classe ricerca

<i>Breve descrizione:</i>	Contiene i criteri per ricercare studenti.
---------------------------	--

<i>Attributi:</i>	<ul style="list-style-type: none"> • id: id dello studente • name: nome studente • surname: cognome studente • classroomId: id della classe
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/15 - SearchTeacherParameterSet - classe ricerca

<i>Breve descrizione:</i>	Contiene i criteri per ricercare professori.
<i>Attributi:</i>	<ul style="list-style-type: none"> • id: id dell'insegnante • subjectId: id della materia insegnata
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/16 - SearchRatingParameterSet - classe ricerca

<i>Breve descrizione:</i>	Contiene i criteri per ricercare le votazioni.
<i>Attributi:</i>	<ul style="list-style-type: none"> • id: id della votazione • studentId: id dello studente che ha ricevuto la votazione • teacherId: id del professore che ha dato il voto • subjectId: id della materia in cui è stato dato il voto
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/17 - SearchAbsenceParameterSet - classe ricerca

<i>Breve descrizione:</i>	Contiene i criteri per ricercare le assenze.
<i>Attributi:</i>	<ul style="list-style-type: none"> • id: id dell'assenza • studentId: id dello studente che ha fatto l'assenza • justified: specifica se l'assenza è giustificata o meno • start: giorno/ora di inizio dell'assenza • end: giorno/ora di fine dell'assenza
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/18 - SearchContactParameterSet - classe ricerca	
<i>Breve descrizione:</i>	Contiene i criteri per ricercare i contatti.
<i>Attributi:</i>	<ul style="list-style-type: none"> • id: id del contatto • personId: id della persona a cui appartengono i contatti • nationId: id della nazione a cui i contatti si riferiscono • city: città
<i>Eredita da:</i>	-
<i>Legata a:</i>	-

CSW-01/19 - DidacticWS - interfaccia del Web Service	
<i>Breve descrizione:</i>	Contiene i metodi che verranno esposti dal Web Service.
<i>Metodi:</i>	<ul style="list-style-type: none"> • getNextPersonId(): ottiene un id usato per creare una nuova persona • addStudent(): aggiunge un nuovo studente • editStudent(): modifica uno studente esistente • deleteStudent(): cancella uno studente con un dato id • searchStudent(): restituisce un insieme di studenti che rispettano i criteri in SearchStudentParameterSet • getNextAbsenceId(): ottiene un id usato per creare una nuova assenza • addAbsence(): aggiunge una nuova assenza associata ad uno studente • editAbsence(): modifica una assenza esistente • deleteAbsence(): cancella una assenza con un dato id • searchAbsence(): restituisce un insieme di assenza che rispettano i criteri in SearchAbsenceParameterSet • getNextRatingId(): restituisce un id usato per creare una nuova valutazione • addRating(): aggiunge una nuova valutazione per uno dato studente in una data materia assegnato da un certo professore • searchRating(): restituisce un insieme di valutazioni che rispettano i criteri in

	<p>SearchRatingParameterSet</p> <ul style="list-style-type: none"> • deleteRating(): cancella una valutazione con un dato id • editRating(): modifica una valutazione esistente • addTeacher(): aggiunge un nuovo professore • editTeacher(): modifica un professore esistente • deleteTeacher(): cancella un professore con un dato id • searchTeacher(): restituisce un insieme di professori che rispettano i criteri in SearchTeacherParameterSet • getNextAbsenceTypeId(): restituisce un id usato per creare una nuova tipologia di assenza • addAbsenceType(): crea una nuova tipologia di assenza • deleteAbsenceType(): cancella una tipologia di assenza • retrieveAllAbsenceTypes(): restituisce tutte le tipologie di assenza • getNextRatingTypeId(): restituisce un id usato per creare un nuovo tipo di valutazione • addRatingType(): crea un nuovo tipo di valutazione • deleteRatingType(): cancella un tipo di valutazione • retrieveAllRatingTypes(): restituisce tutti i tipi di valutazione • getNextSubjectId(): restituisce un id usato per creare una nuova materia • addSubject(): crea una nuova materia • deleteSubject(): cancella una materia • retrieveAllSubjects(): restituisce tutte le materie • getNextClassroomId(): restituisce un id usato per creare una nuova classe • addClassroom(): crea una nuova classe • deleteClassroom(): cancella una classe • retrieveAllClassrooms(): restituisce tutte le classe • retrieveAllNations(): restituisce tutte le nazioni • retrieveAllReligions(): restituisce tutte le religioni • getNextContactId(): ottiene un id usato per creare un nuovo contatto • addContact(): aggiunge un nuovo contatto • editContact(): modifica un contatto esistente • deleteContact(): cancella un contatto con un
--	--

Egoless Group

	<p>dato id</p> <ul style="list-style-type: none">• <code>searchContact()</code>: restituisce un insieme di contatti che rispettano i criteri in <code>SearchContactParameterSet</code>
--	--

4 Stime di fattibilità e di bisogno di risorse

Per l'impegno ed il tempo richiesto alla realizzazione, fare riferimento a [PP].

Per la realizzazione delle componenti, si sono individuate le seguenti tecnologie:

- Java EE 5, in particolare JAX-WS 2.0;
- NetBeans 5.5;
- Sun Java Application Server 9.0 (equivalente a Glassfish)

5 Tracciamento della relazione componenti – requisiti

\	CSA-01 / CSW-01 Didattica	CSA-02 / CSW-02 Documenti	CSA-03 / CSW-03 Monitoraggio	CSA-04 / CSW-04 Accesso telematico
01	x			
02	x			
03	x			
04		x		
05		x		
06			x	
07				
08				x
09		x		
10		x		
11			x	
12	x			
13	x			
14	x			
15				
16		x		
17	x			
18				
19				x
20			x	

\	CSA-01 / CSW-01 Didattica	CSA-02 / CSW-02 Documenti	CSA-03 / CSW-03 Monitoraggio	CSA-04 / CSW-04 Accesso telematico
21	x			
22	x			
23				x
24		x		
25	x			
26			x	
27			x	
28				
29				x
30			x	
31			x	
32	x			