

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Scienze di Internet

S E · R · E D

**Statistical Evaluation and Representation of
Electronic Data**

**Un sistema per l'analisi delle opinioni degli
studenti da integrare con i tool di
valutazione dell'università**

Tesi di Laurea in Programmazione Internet

Relatore:
Chiar.mo Prof.
Antonio Messina

Presentata da:
Andrea Febbo

Correlatore:
Dott. Cristian Mezzetti

Sessione I
Anno Accademico 2011/2012

*A me stesso,
ai miei sforzi,
alla mia perseveranza.*

“Enjoy yourself, it’s later than you think.”

- Herbert A. Magidson -

Prefazione

Tre anni fa, in occasione della mia tesi di laurea, sviluppai un applicativo chiamato SERED. Lo scopo di quest'ultimo era quello di dare uno strumento efficiente ai professori per la valutazione delle opinioni degli studenti. Il software rendeva tutto questo possibile compiendo analisi statistiche e generando grafici sui dati provenienti dai questionari di soddisfazione che gli studenti compilano poco prima di ogni esame.

Allora come oggi l'interpretazione del feedback degli studenti è una parte importante del lavoro dei professori, della segreteria e dell'università tutta. Questo perché senza una accurata conoscenza delle opinioni dei fruitori di un determinato servizio non è possibile capire dove suddetto presenta lacune e dove eccelle. A mio parere questo è un compito importante per ogni somministratore di servizi, non solo quelli di tipo universitario.

Quindi la *raison d'être* di questo applicativo era quella di permettere alla nostra università, con riferimento soprattutto ai professori, di migliorare anno dopo anno i propri servizi didattici.

L'università di Bologna possedeva e possiede strumenti in grado di rendere tutto questo possibile, ma non tutti integrati. Proprio da qui nasce l'idea di sviluppare una nuova versione dell'applicativo, figlia del desiderio di fornire all'ateneo uno strumento migliore, più intuitivo e più integrato di quelli a disposizione.

Il progetto realizzato sarà l'oggetto dello scritto che vi apprestate a leggere.

Andrea Febbo

Indice

Introduzione	xiii
I Realizzazione del Sistema	1
1 Piano di Processo	3
1.1 Visione	3
1.2 Tecnica di Sviluppo Prescelta	4
1.2.1 Descrizione del RUP	4
1.2.2 Piano delle Iterazioni	7
1.3 Artefatti di Stima	8
1.3.1 Diagramma Gantt	8
1.3.2 Diagramma WBS	14
2 Terzo Ciclo di Iterazione	17
2.1 Specifica dei Requisiti	17
2.1.1 Descrizione dei Sottosistemi	19
2.2 Diagrammi UML	26
2.2.1 Descrizione di UML	26
2.2.2 Diagramma dei Casi d'Uso	28
2.2.3 Diagramma di Attività	30
2.2.4 Diagramma di Stato	32
2.2.5 Diagramma di Sequenza	34
2.2.6 Diagramma dei Componenti	39
2.2.7 Diagramma delle Classi	40

2.2.8	Diagramma di Dispiegamento	43
2.3	Scelte Progettuali	43
2.4	Possibili Sviluppi Evolutivi	44
II	Illustrazione del Sistema	47
3	L'Utente Pubblico	49
3.1	Il Grafico Anonimizzato	50
4	L'Utente Comune	53
4.1	Il Grafico non Anonimizzato	54
4.1.1	Overall per Domande	54
4.1.2	Overall per Insegnamenti	54
4.1.3	Singoli Insegnamenti	55
5	L'Utente con Privilegi di ROOT	57
5.1	Afferenze	59
5.2	Utenti	59
5.3	Insegnamenti	60
5.4	Edizioni Insegnamenti	60
5.5	Domande	61
5.6	Struttura Questionario	62
5.7	Questionari	63
5.8	Scarti	63
6	Conclusioni	65
III	Appendici	67
A	Script di creazione del DB	69
A.1	Tabella ANNO	69
A.2	Tabella DOMANDA	69
A.3	Tabella INSEGNAMENTO	70
A.4	Tabella STRUTTURAQUESTIONARIO	71

A.5	Tabella RISPOSTAQUESTIONARIO	71
A.6	Tabella SINGOLOQUESTIONARIO	72
A.7	Tabella SCUOLA_FACOLTA	73
A.8	Tabella MACROAREA	73
A.9	Tabella DIPARTIMENTO	74
A.10	Tabella SEDE	74
A.11	Tabella TIPO	75
A.12	Tabella CDS_DWH_UNIVOCO	75
A.13	Tabella CDS_VARIABILE_ANNUALE	76
A.14	Tabella UTENTE	78
A.15	Tabella DOCENTE_INSEGNAMENTO	78
B	I Tool Utilizzati	81
B.1	Texmaker	81
B.2	Gantt Project	81
B.3	WBS Chart Pro	82
B.4	Poseidon for UML	82
B.5	Eclipse	82
B.6	Hibernate	83
B.7	Shibboleth	83
B.8	Apache Tomcat	84
B.9	Google Chart API	84
B.10	Oracle	84
B.11	PL/SQL Developer	84
B.12	Toad for Oracle	85
B.13	CA ERwin Data Modeler	85
B.14	FastStone Capture	85
C	Tabella di Disambiguazione	87
	Bibliografia	89
	Siti Web Consultati	91

Elenco delle Figure	93
Elenco delle Tabelle	95
Elenco dei Sorgenti	97

Introduzione

La corretta interpretazione dei dati e la conseguente estrazione di informazione da essi è un fattore di primaria importanza per ogni grande istituto. Basti pensare agli enormi database posseduti dalle grandi aziende e dagli enti pubblici nonché allo sforzo che tali entità compiono per analizzare i dati in essi contenuti; per esempio attraverso il data mining.

È certamente vero che l'informazione è potere, ma soltanto se si sa estrarla da i dati in proprio possesso, altrimenti si rischia di perdersi nella loro mole spesso enorme.

Anche la nostra università produce un grande quantità di dati che è interessata ad analizzare. Questo è particolarmente vero in un'area fondamentale come può essere quella riguardante la qualità della didattica, anche perché in base a ciò vengono ripartiti dei fondi.

Estrarre efficacemente informazione dai dati riguardanti quest'area è di primaria importanza poiché questo è di aiuto nel migliorare i servizi didattici e nel semplificare il lavoro dei docenti.

Purtroppo tali dati, a causa delle fonti diverse dalle quali essi provengono, risultano essere frammentati: vi sono dati inerenti alla soddisfazione degli studenti, agli insegnamenti, alla struttura dell'ateneo e molti altri ancora.

Come fare quindi ad estrarre informazione da un insieme frammentato e non integrato di dati? Ovviamente l'unica soluzione è integrarli.

Quest'ultimo è un compito tutt'altro che banale poiché, se non esplicito correttamente, comporta la perdita di coerenza fra i dati stessi, andando quindi a scapito di tale estrazione.

Si rende quindi necessario, per un più corretto ed efficace utilizzo dei dati in questione, uno strumento che li misceli sapientemente in un'unica struttura

coerente e che possa estrarre tutta l'informazione che essi contengono.

Tale sistema dovrebbe dare anche la possibilità di condividere pubblicamente l'informazione estratta, cosicché anche gli studenti possano verificare a quale livello qualitativo si trovano i loro corsi di studio.

Per le ragioni sopraelencate si è ritenuto utile sviluppare lo strumento in questione; scopo di questo documento è di descriverlo.

La presentazione è suddivisa in tre parti: la prima descrive il lavoro svolto dal punto di vista della realizzazione, la seconda descrive l'applicativo realizzato e la terza contiene le appendici.

Parte I

Realizzazione del Sistema

Capitolo 1

Piano di Processo

1.1 Visione

Si intende sviluppare una nuova versione, per la precisione la terza, del software denominato SERED. Questa versione dovrà possedere le seguenti funzionalità:

- un'efficiente importazione dei dati;
- la gestione di grandi quantità di essi;
- significative interpretazioni grafiche;
- grande granularità nella scelta dei dati da analizzare;
- un'efficace sistema di storage che permetta la creazione di uno storico.
- integrarsi con gli altri servizi sui server dell'università.
- essere realizzato con lo stack tecnologico richiesto dall'università per i propri sistemi;
- girare sul web in modo da rendere possibile la fruizione a tutti coloro che ne avessero il desiderio.

Una delle sfide più avvincenti di questo progetto viene sicuramente dalla non uniformità dei dati che si andranno a manipolare. Questa non uniformità

è dovuta al fatto che tali dati provengono da fonti differenti: alcuni di essi vengono dal CeSIA, altri dalla segreteria ed altri ancora sono il risultato della lettura dei questionari da parte di un altro gruppo attraverso l'utilizzo di un lettore ottico. Cercare di miscelare dati di origini così diverse non è certo un lavoro da poco.

Un altro fattore che aumenta la complessità del progetto è l'estrema variabilità della struttura dei dati. Ad esempio il sistema dovrà gestire situazioni come il variare delle afferenze degli istituti dell'università fra loro o il variare del numero di domande dei questionari nel corso degli anni.

1.2 Tecnica di Sviluppo Prescelta

Per quanto riguarda la tecnica di sviluppo si è scelto di continuare ad usare il metodo iterativo ispirato al RUP (Rational Unified Process) con il quale erano state sviluppate le prime due versioni del software.

In questa prospettiva la terza versione verrà vista anche come il terzo ciclo di iterazione, considerando anche la somiglianza nei requisiti funzionali.

1.2.1 Descrizione del RUP

Il Rational Unified Process (RUP) è un modello di processo software iterativo sviluppato da Rational Software pensato soprattutto per progetti di grandi dimensioni. I creatori del RUP partirono dalla diagnosi di un campione di progetti software falliti, allo scopo di identificare cause tipiche o generali di fallimento. Quindi confrontarono questa informazione con la struttura dei processi software descritti in letteratura e applicati nella pratica, cercando di identificare le soluzioni proposte precedentemente. L'elenco dei motivi di fallimento identificati comprende per esempio:

- Gestione ad hoc (non standard) dei requisiti;
- Comunicazione ambigua e non precisa;
- Architettura fragile (incapace di sopportare situazioni di particolare criticità) ;

- Incapacità di gestire la complessità;
- Inconsistenze nei requisiti, nel progetto o nelle implementazioni;
- Collaudo insufficiente;
- Valutazione soggettiva dello stato del processo;
- Incapacità di affrontare il rischio;
- Propagazione non controllata delle modifiche;
- Insufficiente automazione.

Il RUP si può descrivere come una collezione di best practices mirate a evitare questi e altri problemi, e come un ambiente di gestione dei processi che facilita l'applicazione di tali pratiche. Il processo fu progettato utilizzando strumenti tipici della progettazione del software; in particolare, esso fu descritto in termini di un metamodello object-oriented, espresso in UML. Nel RUP, il ciclo di vita di un processo software viene suddiviso in cicli di sviluppo, a loro volta scomposti in fasi. Le fasi previste sono:

- Inception Phase (fase iniziale);
- Elaboration Phase (fase di elaborazione);
- Construction Phase (fase di costruzione);
- Transition Phase (fase di transizione).

Ogni fase ha un certo insieme di obiettivi e si conclude con la realizzazione di un deliverable (prodotto) di qualche genere. Le fasi sono ulteriormente scomposte in iterazioni, che sono associate a periodi temporali e hanno deadline precise.

L'Inception Phase si può considerare come una particolare elaborazione e precisazione del concetto generale di analisi di fattibilità. Lo scopo principale è quello di delineare nel modo più accurato possibile il business case, ovvero comprendere il tipo di mercato al quale il progetto afferisce e identificare gli

elementi importanti affinché esso conduca a un successo commerciale. Fra gli strumenti utilizzati ci sono un modello dei casi d'uso, la pianificazione iniziale del progetto, la valutazione dei rischi, una definizione grossolana dei requisiti e così via. Se il progetto non supera questa milestone, detta "Lifecycle Objective Milestone", esso dovrà essere abbandonato o ridefinito.

La fase di elaborazione definisce la struttura complessiva del sistema. Questa fase comprende l'analisi di dominio e un prima fase di progettazione dell'architettura. Questa fase deve concludersi con il superamento di una milestone detta "Lifecycle Architecture Milestone". A questo scopo devono essere soddisfatti i seguenti criteri:

- deve essere stato sviluppato un modello dei casi d'uso completo all'80%;
- dev'essere fornita la descrizione dell'architettura del sistema;
- dev'essere stata sviluppata un "architettura eseguibile" che dimostra il completamento degli use case significativi;
- dev'essere eseguita una revisione del business case e dei rischi;
- dev'essere completata una pianificazione del progetto complessivo.

Se il progetto non passa questa milestone, potrebbe ancora essere abbandonato, oppure dovrà essere rivisitato. Al termine di questa fase si transita infatti in una situazione di rischio più elevato, in cui le modifiche all'impostazione del progetto saranno più difficili e dannose.

Nella fase Construction Phase viene portato a termine il grosso degli sviluppi. Viene prodotta la prima release del sistema. La milestone di questa fase si chiama "Initial Operational Capability" e rappresenta la prima disponibilità delle funzionalità del sistema in termini di implementazione.

Nella fase di transizione, il sistema passa dall'ambiente dello sviluppo a quello del cliente finale. Vengono condotte le attività di training degli utenti e il beta testing del sistema a scopo di verifica e validazione. Si deve in particolare verificare che il prodotto sia conforme alle aspettative descritte nella fase di Inception. Se questo non è vero si procede a ripetere l'intero ciclo;

altrimenti, si raggiunge la milestone detta “Product Release” e lo sviluppo termina.

Tipicamente, un progetto gestito usando il RUP viene suddiviso in iterazioni. Questa scomposizione presenta numerosi vantaggi (in particolare rispetto alla valutazione dell’avanzamento del progetto e alla gestione dei fattori di rischio) ma implica un overhead specifico. Il RUP definisce una “Project Management Discipline” (disciplina di gestione dei progetti) a cui il project manager può affidarsi per amministrare le iterazioni.

Per informazioni aggiuntive vedi **Bibliografia [4]**.

1.2.2 Piano delle Iterazioni

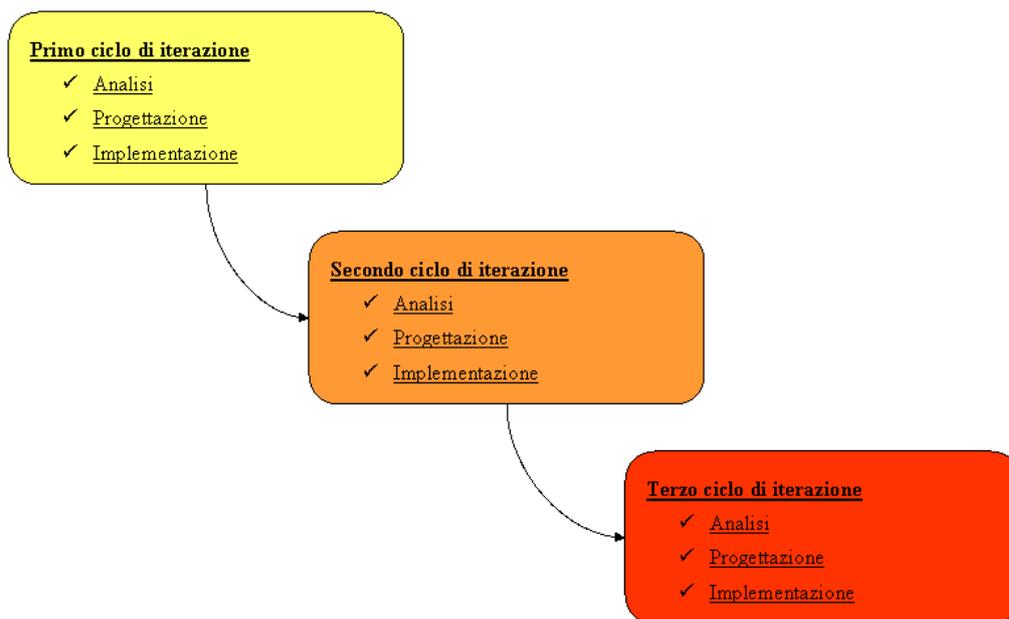


Figura 1.1: Il diagramma delle iterazioni

È stato previsto l’uso di una iterazione per la creazione della terza versione del software. Si è ritenuto che una iterazione fosse ampiamente sufficiente a garantire il raggiungimento degli obiettivi. Questo poiché nelle prime due iterazioni il progetto era stato impostato per facilitare l’inserimento di

funzioni e la conseguente creazione di nuove versioni con l'aggiunta di altri cicli d'iterazione, facilitando così la manutenzione adattiva e quella perfetta.

1.3 Artefatti di Stima

Una volta scelta la tecnica di sviluppo da utilizzare, in questo caso quella iterativa, è fondamentale realizzare una previsione della tempistiche e delle risorse necessarie ad ogni attività.

La scelta dei documenti da produrre è cruciale per l'ottimale riuscita di un progetto di tale portata, soprattutto per quanto riguarda la stima. I documenti che infine si è scelto di produrre per le stime sono:

- Stima delle tempistiche con il diagramma Gantt (vedi **Capitolo 1.3.1**);
- Stima delle attività con il diagramma WBS (vedi **Capitolo 1.3.2**).

1.3.1 Diagramma Gantt

Descrizione dei Diagrammi Gantt

I diagrammi di Gantt sono degli strumenti di supporto alla gestione dei progetti, così chiamati in ricordo dell'ingegnere statunitense Henry Laurence Gantt (1861 - 1919), studioso di scienze sociali, che li ideò nel 1917.

Usati principalmente nelle attività di project management, i diagrammi sono costruiti partendo da un asse orizzontale - a rappresentazione dell'arco temporale totale del progetto, suddiviso in fasi incrementalmente (ad esempio, giorni, settimane, mesi) - e da un asse verticale - a rappresentazione delle mansioni o attività che costituiscono il progetto.

Barre orizzontali di lunghezza variabile rappresentano le sequenze, la durata e l'arco temporale di ogni singola attività del progetto (l'insieme di tutte le attività del progetto ne costituisce la Work Breakdown Structure (vedi **Capitolo 1.3.2**)). Queste barre possono sovrapporsi durante il medesimo arco temporale ad indicare la possibilità dello svolgimento in parallelo di alcune delle attività. Man mano che il progetto progredisce, delle barre secondarie, delle frecce o delle barre colorate potranno essere aggiunte al diagramma,

per indicare le attività sottostanti completate o una porzione completata di queste. Una linea verticale è utilizzata per indicare la data di riferimento.

Un diagramma di Gantt permette dunque la rappresentazione grafica di un calendario di attività, utile al fine di pianificare, coordinare e tracciare specifiche attività in un progetto, dando una chiara illustrazione dello stato d'avanzamento del progetto rappresentato; di contro, uno degli aspetti non tenuti in considerazione in questo tipo di diagramma è l'interdipendenza delle attività, caratteristica invece della programmazione reticolare, cioè del diagramma Pert. Ad ogni attività possono essere in generale associati una serie di attributi: durata (o data di inizio e fine), predecessori, risorsa, costo.

Ad ogni attività possono essere associate una o più risorse. Alcuni software disponibili sul mercato permettono di visualizzare il carico di lavoro di ogni risorsa e la sua saturazione, impostando una certa disponibilità per ogni risorsa. Contestualmente, può essere definito il calendario dei giorni lavorativi e festivi, e il numero di ore di lavoro giornaliera.

Ad ogni attività può inoltre essere associato un costo. Il costo può essere attribuito a una singola attività oppure si può assegnare un costo orario alle risorse, determinando il costo dell'attività in base al relativo impegno orario. Nel corso del progetto, ad ogni attività o alla risorsa può essere attribuito un costo effettivo. Dai due dati di costo, preventivati al momento della stesura del Gantt, ed effettivi, si ricavano tre curve e due indicatori di avanzamento dell'intero progetto. Le tre curve riportano la cumulata del costo preventivato e/o effettivo in funzione del tempo, ossia i costi totali effettivi e/o preventivati dall'inizio:

- BCWS: Budget Cost of Work Scheduled: riporta i costi preventivati da 0 a vita intera (da 0 a $t = T$);
- ACWP: Actual Cost of Work Performed: riporta i costi sostenuti da 0 a tempo attuale (da 0 a $t = T_0$);
- BCWP. Budget Cost of Work Performed: riporta i costi preventivati da 0 a tempo attuale (da 0 a $t = T_0$).

Diagramma Gantt del Progetto

Per generare il diagramma Gantt si è deciso di suddividere l'intero processo di sviluppo nelle seguenti macro-sezioni:

1. piano di processo;
2. ciclo di iterazione;
 - (a) analisi;
 - (b) progettazione;
 - (c) implementazione;
3. documentazione.

Per creare il diagramma Gantt del progetto è stato usato, come risulterà dall'elenco dei tool utilizzati (vedi **Capitolo B.2**), Gantt Project. Qui di seguito è allegata la documentazione generata con il suddetto programma, completa di tutte le tempistiche e naturalmente del diagramma Gantt definitivo.

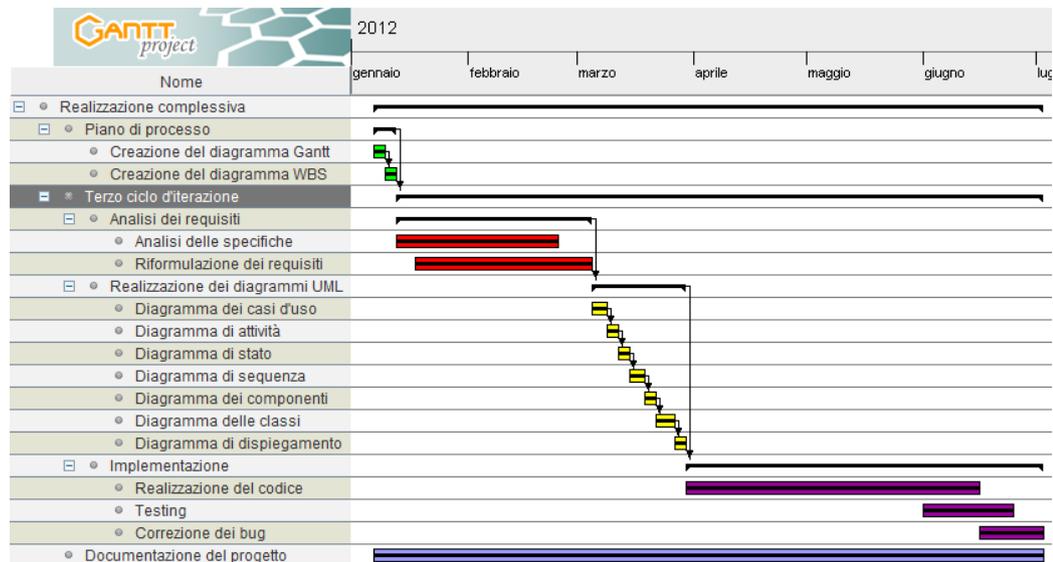


Figura 1.2: Il diagramma Gantt del progetto

Lista attività				
Nome	Data		Risorse	
	Data d'inizio	Data di fine		
Realizzazione complessiva	07/01/12	03/07/12		
Piano di processo	07/01/12	13/01/12		
Creazione del diagramma Gantt	07/01/12	10/01/12	Andrea Febbo	
Creazione del diagramma WBS	10/01/12	13/01/12	Andrea Febbo	
Terzo ciclo d'iterazione	13/01/12	03/07/12		
Analisi dei requisiti	13/01/12	05/03/12		
Analisi delle specifiche	13/01/12	25/02/12	Andrea Febbo	
Riformulazione dei requisiti	18/01/12	05/03/12	Andrea Febbo	
Realizzazione dei diagrammi UML	05/03/12	30/03/12		
Diagramma dei casi d'uso	05/03/12	09/03/12	Andrea Febbo	
Diagramma di attività	09/03/12	12/03/12	Andrea Febbo	
Diagramma di stato	12/03/12	15/03/12	Andrea Febbo	
Diagramma di sequenza	15/03/12	19/03/12	Andrea Febbo	
Diagramma dei componenti	19/03/12	22/03/12	Andrea Febbo	
Diagramma delle classi	22/03/12	27/03/12	Andrea Febbo	
Diagramma di dispiegamento	27/03/12	30/03/12	Andrea Febbo	
Implementazione	30/03/12	03/07/12		
Realizzazione del codice	30/03/12	16/06/12	Andrea Febbo	
Testing	01/06/12	25/06/12	Andrea Febbo	
Correzione dei bug	16/06/12	03/07/12	Andrea Febbo	
Documentazione del progetto	07/01/12	03/07/12	Andrea Febbo	

Figura 1.3: La lista delle attività stimate con Gantt Project

Per stimare le tempistiche di questo progetto ci si è affidati all'esperienza accumulata negli altri progetti di sviluppo software svolti durante il corso degli studi universitari; ciò ha portato senz'altro a buoni risultati dato che sono state rispettate fundamentalmente tutte le scadenze previste.

Descrizione del Piano di Processo

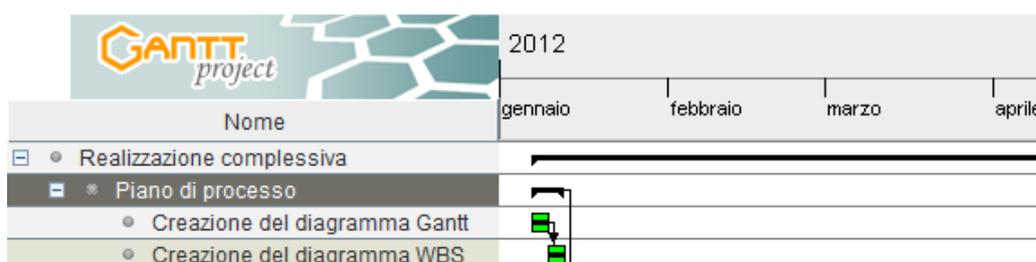


Figura 1.4: Il piano di processo nel diagramma Gantt

La prima attività svolta è stata la creazione del piano di processo. È stato stimato che questa attività avrebbe richiesto sei giorni di lavoro. Le due sotto-attività del piano di processo sono rispettivamente:

- la creazione del diagramma Gantt;
- la creazione del diagramma WBS.

Si è ritenuto che tre giorni di lavoro per ciascuna di queste operazioni sarebbero stati sufficienti. Subito dopo l'attività sopraccitata, ha avuto inizio il terzo ciclo iterativo di sviluppo.

Descrizione del Terzo Ciclo di Iterazione

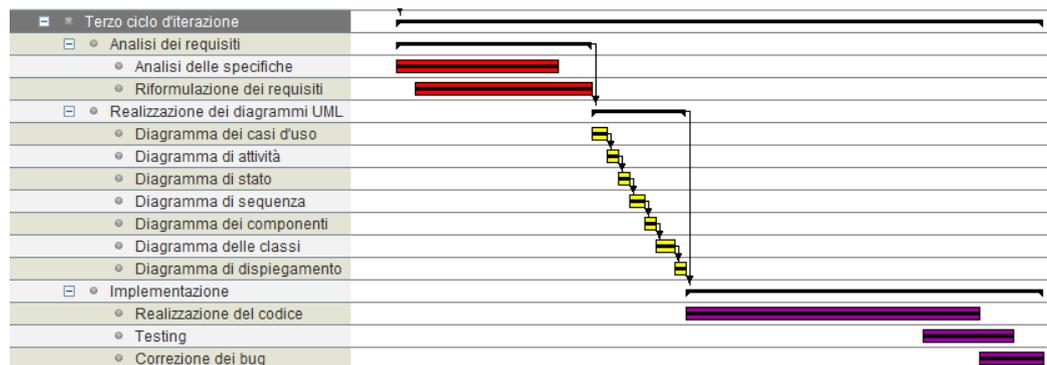


Figura 1.5: Il terzo ciclo di iterazione nel diagramma Gantt

Essendo questa attività di una complessità non banale si è ritenuto opportuno assegnarle un quanto di tempo commisurato a tale complessità: centotettantadue giorni. Il primo ciclo di sviluppo si divide a sua volta in tre importanti sotto-attività:

- l'analisi dei requisiti;
- la realizzazione dei diagrammi UML;
- l'implementazione vera e propria del programma.

L'analisi dei requisiti è costituita dall'analisi delle specifiche e dalla conseguente riformulazione dei requisiti stessi. Data la complessità intrinseca delle specifiche, questa attività ha richiesto cinquantadue giorni.

Una volta terminata l'analisi dei requisiti si è proceduto alla realizzazione dei diagrammi UML, pianificando un'attività di venticinque giorni. I diagrammi sviluppati sono:

- il diagramma dei casi d'uso;
- il diagramma di attività;
- il diagramma di stato;
- il diagramma di sequenza;
- il diagramma dei componenti;
- il diagramma delle classi;
- il diagramma di dispiegamento.

Ad ognuno di questi diagrammi, tranne a quello delle classi, è stato assegnato un solo giorno di lavoro in fase di stima.

Ai diagrammi più semplici (attività, stato, componenti e dispiegamento) sono stati assegnati tre giorni. A quelli mediamente complessi (casi d'uso e sequenza) sono invece stati assegnati quattro giorni. Al diagramma delle classi, che è sicuramente il più complesso da realizzare sono stati assegnati cinque giorni.

Arrivati a questo punto possiamo trovare come ultima sotto-attività del primo ciclo di iterazione la fase di implementazione. Questa fase è suddivisa in tre operazioni, ovvero:

- realizzazione del codice;
- testing;
- correzione dei bug.

È stato previsto che la fase di implementazione venga effettuata in novantacinque giorni.

Descrizione della Fase di Documentazione del Progetto



Figura 1.6: La fase di documentazione del progetto nel diagramma Gantt

Questo progetto prevede una fase di documentazione da effettuarsi durante tutta la sua durata. La fase di documentazione prevede di la descrizione analitica e la chiara esplicazione di tutte le operazioni presenti dall'inizio alla fine del progetto.

1.3.2 Diagramma WBS

Descrizione dei Diagrammi WBS

Con l'espressione inglese Work Breakdown Structure (WBS, Struttura Analitica di Progetto) si intende l'elenco di tutte le attività di un progetto. Le WBS sono usate nella pratica del Project management e coadiuvano il project manager nell'organizzazione delle attività di cui è responsabile.

Molto spesso i progetti sono composti da migliaia di attività: per facilitare il lavoro di organizzazione delle varie attività esistono delle WBS-tipo che elencano tutte le possibili attività (generiche) per i progetti del rispettivo ambito. L'insieme delle attività può quindi essere confrontata con una checklist.

Diagramma WBS del Progetto

Per ottenere un diagramma¹ chiaro e conciso si è scelto di includere in esso le attività prese dal punto di vista del risultato senza entrare nel merito di come tali risultati verranno ottenuti, quindi indicando solo cosa dovrà essere fatto. Si è inoltre scelto di omettere dati come le tempistiche, visto che sono ampiamente descritte nel diagramma Gantt, o il costo, dato che in questo caso il progetto non è stato svolto sotto retribuzione.

Nel diagramma prodotto per questo progetto è presente, su ogni nodo dell'albero, la percentuale di avanzamento delle attività. Questa è una scelta

¹Diagramma ottenuto con WBS Chart Pro (vedi **Capitolo B.3**)

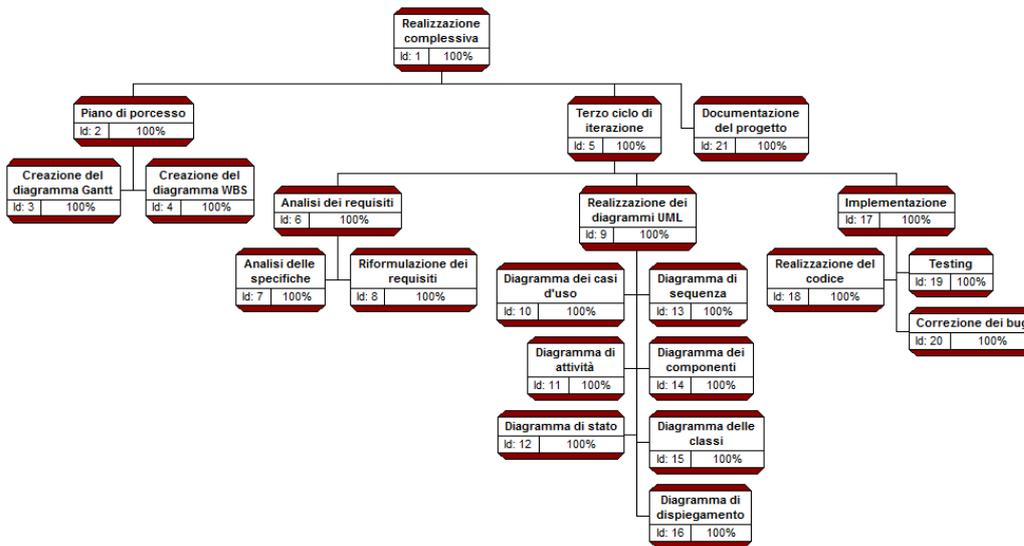


Figura 1.7: Il diagramma WBS del progetto

che è stata presa per permettere il più alto livello di monitoraggio possibile durante ogni fase del progetto. Ovviamente il diagramma WBS mostrato si riferisce al progetto finito.

Capitolo 2

Terzo Ciclo di Iterazione

2.1 Specifica dei Requisiti

Si dovrà realizzare un sistema basato sul web che permetta di raccogliere, visualizzare e analizzare i dati corrispondenti alle schede di valutazione che gli studenti periodicamente compilano. Per far sì che ciò avvenga, tale sistema avrà bisogno di una enorme mole di dati che modelleranno la struttura dell'università tutta.

Il sistema dovrà essere flessibile a ogni cambiamento che i dati potranno presentare nel tempo, sia quelli riguardanti la struttura stessa dell'università (un CdS¹ che cambia Scuola), sia il numero e la collocazione delle domande nei questionari.

Per garantire le seguenti caratteristiche è ovvio che ci sarà bisogno di utilizzare un DBMS. Il DB dovrà essere modellato con una struttura che permetta di gestire uno storico dei dati in modo da poterli valutare nella loro evoluzione temporale.

Verranno gestiti dal sistema tre livelli di utenza.

Utente pubblico: sono tutti coloro che sono interessati alla qualità dell'insegnamento dell'università. Costoro non avranno bisogno di alcun log in poiché avranno accesso solo ai dati pubblici in forma anonima e raggruppati per CdS.

¹Per gli acronimi consultare la tabella di disambiguazione (**Tabella C.1**).

Utente comune: sono fondamentalmente i docenti. Questi ultimi potranno avere accesso ai dati relativi ai loro insegnamenti, oltre che naturalmente a quelli pubblici.

Utente con privilegi di ROOT: è quell'utente che si occuperà della gestione ordinaria del sistema, ovvero di caricare di anno in anno il DB con i relativi dati. Talvolta ci si potrà riferire a questo utente semplicemente come utente di ROOT.

Il sistema permetterà: a chiunque ne sia interessato (utente pubblico) la visualizzazione di dati in forma anonima e aggregati per corsi, ad ogni docente di consultare nello specifico i dati che a lui fanno riferimento (utente comune) ed al manutentore del sistema (utente con privilegi di ROOT) di caricare i dati.

Dato che questo sistema dovrà essere integrato sui server dell'università in modo che i suoi servizi possano essere utilizzati da docenti, studenti, e chiunque ne sia interessato, l'autenticazione dovrà avvenire con il single sign-on (SSO) come gli altri sistemi messi a disposizione dall'ateneo. Il servizio di SSO universitario viene offerto tramite Shibboleth. Quest'ultimo è un sistema open-source che offre funzionalità di single sign-on e permette ai siti di prendere decisioni informate sull'autorizzazione per l'accesso individuale a risorse on-line protette in un ottica di preservazione della privacy. Per maggiori informazioni consultare l'**Appendice B.7**.

Conseguentemente alla sopraccitata integrazione il sistema dovrà essere sviluppato con lo stack tecnologico usato dalle altre applicazioni residenti sui server universitari. Il suddetto stack prevede le seguenti tecnologie:

- Apache Tomcat;
- JSP;
- Hibernate;
- Oracle 11g;
- Shibboleth.

Gestione della Struttura dell'Università Nel modellare la struttura dell'università si partirà dal CdS poiché gli stakeholder lo hanno definito come il nucleo di tale struttura.

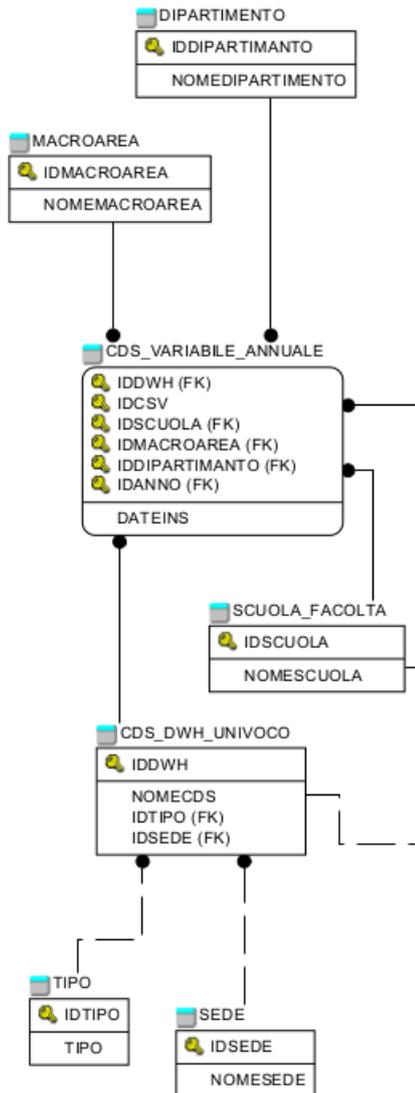


Figura 2.2: Le tabelle relative alla gestione della struttura dell'università

Si avrà bisogno di due tabelle per modellare i CdS, CDS_DWH_UNIVOCO e CDS_VARIABLEI_ANNUALE, questo perché vi sono delle informazioni sui corsi che non cambiano mai ed altre che mutano di anno in anno.

La prima tabella contiene tutte quelle informazioni che non variano mai: il nome del CdS, l'ID del tipo (laurea triennale, magistrale, etc) e l'ID della sede. La Primary key di questa tabella è l'IDDDWH: un ID fornito dal data warehouse che identifica univocamente i CdS.

La seconda tabella, al contrario, contiene tutte le informazioni che variano di anno in anno: gli ID della scuola, della macroarea, del dipartimento, dell'anno e l'IDCSV. Abbiamo bisogno di sapere che IDCSV ha un CdS di anno in anno poiché nei questionari ci si riferisce a quest'ultimo usando appunto suddetto ID. Questo avviene poiché i dati che andranno a popolare il nostro DB provengono da fonti diverse.

Questo meccanismo ci permette di “fotografare” lo stato dell'università ed in particolare dei CdS di anno in anno permettendoci così di capire, ad esempio, che il corso di studio XYZ nell'anno N afferisce ad una determinata scuola, ad un determinato dipartimento, ad una data macroarea ed ha un determinato codice CSV

(Indispensabile per importare i dati).

Vi sono poi ovviamente tutte le tabelle che forniscono le foreign key alle due dei CdS: SCUOLA_FACOLTA, DIPARTIMENTO, MACROAREA, TIPO e SEDE.

Gestione dei Questionari Per gestire le variazioni che la struttura dei questionari subisce di anno in anno si utilizzerà la tabella denominata STRUTTURAQUESTIONARIO che ci permetterà di capire in che posizione si trova ogni domanda al variare degli anni e su che scala verrà valutata. Il campo scala ci permetterà di calcolare il livello di soddisfazione in percentuale. Questa tabella ha una primary key sintetica denominata IDSTRUTTURAQUESTIONARIO.

Una volta modellata la struttura del questionario si rende necessario fare lo stesso con in singoli questionari. A questo scopo verranno create le tabelle SINGOLOQUESTIONARIO e RISPOSTAQUESTIONARIO.

La prima tabella conterrà tutte le informazioni relative al questionario in questione: L'ID della scuola, del corso, dell'insegnamento e l'anno di iscrizione dello studente che compila. Questa tabella ha una primary key sintetica denominata IDSINGOLOQUESTIONARIO e un campo contenete la data di inserimento del record denominato DATEINS.

La seconda tabella, invece, conterrà le informazioni relative alle risposte dei questionari: a quale questionario si riferisce (IDSINGOLOQUESTIONARIO), di quale domanda è la risposta (IDSTRUTTURAQUESTIONARIO) e la risposta in se. Anche questa tabella ha una primary key sintetica (IDRI-

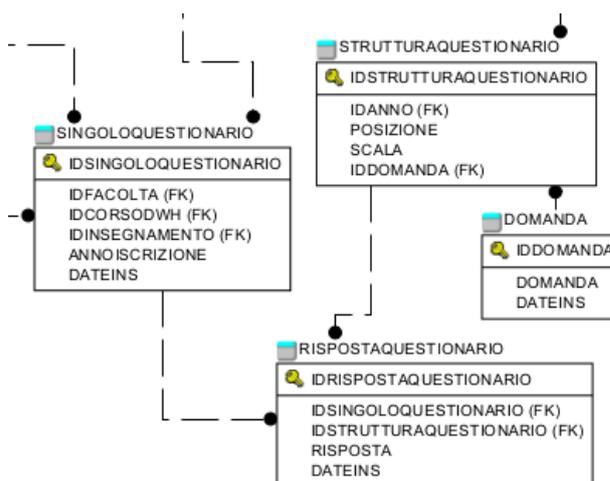


Figura 2.3: Le tabelle relative alla gestione dei questionari

SPOSTAQUESTIONARIO) e un campo contenete la data di inserimento del record (DATEINS).

Gestione degli Insegnamenti Come abbiamo visto in precedenza ogni questionario compilato si riferisce ad un insegnamento.

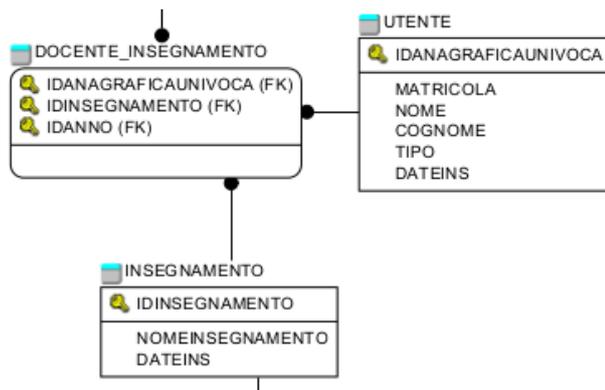


Figura 2.4: Le tabelle relative alla gestione degli insegnamenti

Tutti gli insegnamenti, come è ovvio, verranno inseriti nella tabella INSEGNAMENTO. La primary key di questa tabella è l'IDINSEGNAMENTO che è il codice che normalmente l'università usa per identificare gli insegnamenti. Sono presenti anche il nome dell'insegnamento e la data di inserimento del record.

La tabella UTENTE contiene tutti i dati relativi ai docenti ed a ogni altro utente che potrà fare log in nel sistema. La primary key di questa tabella è l'IDANAGRAFICAUNIVOCA che è un identificativo univoco che l'università ha di tutti coloro che hanno mai avuto qualche genere di rapporto con essa. Gli altri campi della tabella sono la MATRICOLA, il NOME, il COGNOME, il TIPO e la data di inserimento del record. L'attributo tipo servirà a distinguere fra l'utente comune e quello con privilegi di ROOT.

Abbiamo la tabella denominata DOCENTE_INSEGNAMENTO che serve a capire ogni edizione di un dato insegnamento da quale docente è tenuta.

Considerando la frammentarietà dei dati di partenza e la loro estrema variabilità, nonché ambiguità nel tempo, questa soluzione è sicuramente la migliore alla quale si poteva arrivare.

Il Sottosistema di Acquisizione

Stabilita la struttura del DB si dovrà creare un sistema di acquisizione che dovrà popolare il DB steso. Esisteranno tre modalità di popolamento delle

tabelle che ora andremo ad elencare.

Popolamento da script: per popolare alcune tabelle, quelle che dovranno essere popolate una tantum, verranno usati dei semplici script SQL. Queste tabelle sono:

- CDS_DWH_UNIVOCO;
- TIPO;
- SEDE;
- SCUOLA_FACOLTA;
- MACROAREA;
- DIPARTIMENTO.

Popolamento da file: le tabelle che dovranno essere caricate periodicamente e che dovranno contenere una grande mole di record verranno popolate tramite delle funzioni software che leggeranno dei file CSV ed inseriranno i dati di conseguenza. Queste tabelle sono:

- CDS_VARIABILE_ANNUALE;
- ANNO²;
- DOCENTE_INSEGNAMENTO;
- UTENTE³;
- INSEGNAMENTO;
- SINGOLOQUESTIONARIO;
- RISPOSTAQUESTIONARIO.

²Questa tabella non conterrà una grande quantità di record ma si è ritenuto opportuno far sì che venga caricata dalla funzione che popolerà la tabella CDS_VARIABILE_ANNUALE cosicché un nuovo anno verrà “creato” nel DB quando verrà “fotografata” una nuova struttura dell’università.

³La tabella UTENTE sarà popolata da file tranne per il primo record: l’inserimento del primo utente con privilegi di ROOT (vedi **Sezione 5.2**)

Popolamento manuale: le tabelle che dovranno essere caricate periodicamente ma che avranno quantità infime di record saranno popolate tramite funzioni di inserimento manuale. Queste tabelle sono:

- DOMANDA;
- STRUTTURAQUESTIONARIO;

Per quanto riguarda il popolamento delle tabelle tramite file è utile capire quali tipi di file popoleranno quali tabelle: andiamo quindi a descriverli.

Afferenze popola le tabelle:

- CDS_VARIABLEE_ANNUALE;
- ANNO.

Utenti popola la tabella:

- UTENTE.

Insegnamenti popola la tabella:

- INSEGNAMENTO.

Edizioni insegnamenti popola la tabella:

- DOCENTE_INSEGNAMENTO;

Questionari popola le tabelle:

- SINGOLOQUESTIONARIO;
- RISPOSTAQUESTIONARIO.

Il sottosistema di acquisizione dovrà inserire i record nelle tabelle facendo in modo che l'integrità dei dati venga mantenuta a tutti i costi, nonostante il variare di questi ultimi. Ad esempio, per quanto riguarda il popolamento della tabella SINGOLOQUESTIONARIO, il sistema dovrà capire

a quale IDWH si riferiscono i codici IDCSV e IDSCUOLA nell'anno per il quale si sta caricando, così da poter inserire un record con un identificativo univoco (IDWH) piuttosto che uno variabile (IDCSV e IDSCUOLA). Questa operazione verrà effettuata attraverso delle select alla tabella CDS_VARIABILE_ANNUALE prima di inserire il record in SINGOLOQUESTIONARIO.

Il Sottosistema GUI

Il sistema avrà un interfaccia utente intuitiva ed user friendly che prevederà tre insiemi di funzioni quanti sono i tipi di utenti che prenderemo in considerazione.

Andiamo ora a riassumere, tramite la **Tabella 2.1**, a quali di questi insiemi di funzioni ogni tipo di utente avrà accesso.

Tipo di utente	Insiemi di funzioni		
	Caricamento dati	Consultazione dati privati	Consultazione dati pubblici
Utente di ROOT	✓	✓	✓
Utente comune		✓	✓
Utente pubblico			✓

Tabella 2.1: Tabella delle funzioni

Ogni insieme di funzioni, oltre naturalmente a gestire la business logic retrostante le funzioni stesse, dovrà offrire soluzioni grafiche intuitive per la fruizione di queste ultime.

Caricamento dati: questo insieme di funzioni offrirà, dal punto di vista dell'interfaccia utente, oltre alle singole pagine che permetteranno il caricamento dei dati, una pagina di riepilogo così da far capire con facilità quali tabelle sono state caricate e quali devono ancora esserlo.

Consultazione dati privati: questo insieme di funzioni offrirà ad ogni docente un prospetto contenente tutti gli insegnamenti tenuti nel corso

degli anni. Sarà quindi possibile generare grafici su un singolo insegnamento o su tutti gli insegnamenti di un anno. Questi grafici conterranno la media delle risposte alle varie domande sugli insegnamenti selezionati.

Consultazione dati pubblici: questo insieme di funzioni offrirà, su ogni CdS, la possibilità di visualizzare un grafico contenente la media delle risposte alle varie domande sugli insegnamenti che compongono il CdS selezionato. Per la tutela della privacy dei docenti gli insegnamenti contenuti in suddetto grafico saranno anonimizzati.

2.2 Diagrammi UML

L'uso di UML è molto importante in ogni progetto software che si rispetti, affinché attraverso una modellazione ben precisa si possa raggiungere la più grande eleganza architeturale possibile. Questa caratteristica è fondamentale perché tra le altre cose, permette una facile manutenzione del programma.

Potrà capitare che un aspetto del programma venga descritto in più diagrammi. Questa peculiarità non è frutto di ridondanza bensì di una accurata descrizione di ogni elemento dal punto di vista dei vari diagrammi.

Detto ciò mi accingo a fornire una breve descrizione di UML.

2.2.1 Descrizione di UML

In ingegneria del software, UML (Unified Modeling Language, “linguaggio di modellazione unificato”) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. Il nucleo del linguaggio fu definito nel 1996 da Grady Booch, Jim Rumbaugh e Ivar Jacobson (detti “i tre amigos”) sotto l'egida dello OMG⁴, che tuttora gestisce lo standard di UML. Il linguaggio nacque con l'intento di unificare approcci precedenti (dovuti ai tre padri di

⁴L'Object Management Group (OMG) è un consorzio creato nel 1989 con 440 aziende tra cui: Microsoft, Digital, HP, NCR, SUN, OSF con l'obiettivo di creare un sistema di gestione di un'architettura distribuita.

UML e altri), raccogliendo le best practices nel settore e definendo così uno standard industriale unificato.

UML svolge un'importantissima funzione di lingua franca nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa UML per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico.

La modellazione UML avviene principalmente attraverso diagrammi che sono divisi in due categorie, come si vede in **Figura 2.5**.

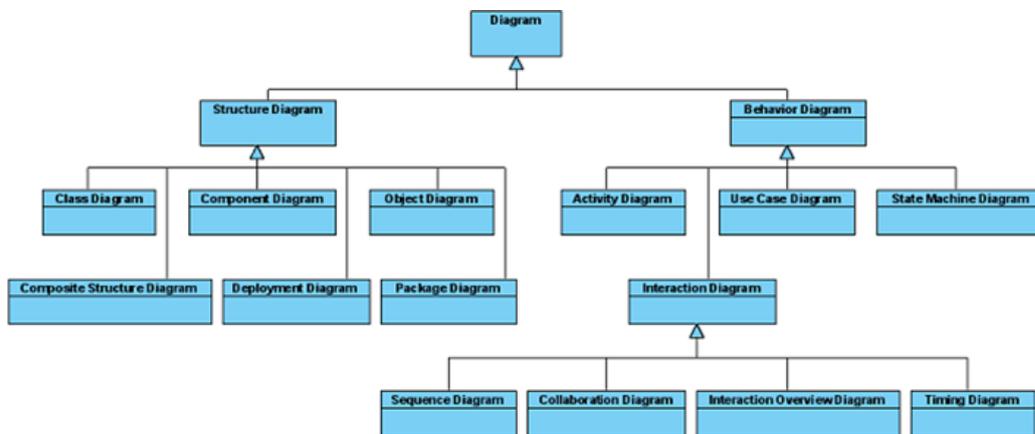


Figura 2.5: Classificazione dei diagrammi UML

Andremo ora a descrivere tali categorie.

Diagrammi di struttura: descrivono le strutture statiche dei dati, e si possono ottenere utilizzando per esempio i diagrammi delle classi, i diagrammi di dispiegamento o i diagrammi dei componenti.

Diagrammi di comportamento: descrivono la collaborazione tra oggetti. Ci sono molte tecniche di visualizzazione per la modellazione del comportamento, come il diagramma dei casi d'uso, il diagramma di sequenza, il diagramma di stato, il diagramma di collaborazione e il diagramma di attività.

L'ultima versione del linguaggio, la 2.0, è stata consolidata nel 2004 e ufficializzata da OMG nel 2005. UML 2.0 riorganizza molti degli elementi della

versione precedente (1.5) in un quadro di riferimento ampliato e introduce molti nuovi strumenti, inclusi alcuni nuovi tipi di diagrammi. Sebbene OMG indichi UML 2.0 come la versione “corrente” del linguaggio, la transizione è di fatto ancora in corso; le stesse specifiche pubblicate da OMG sono ancora non completamente aggiornate e il supporto dei tool a UML 2.0 è, nella maggior parte dei casi, appena abbozzato.

Per informazioni aggiuntive si consulti Wikipedia (**Siti Web Consultati [16]**).

2.2.2 Diagramma dei Casi d’Uso

Descrizione dei Diagrammi dei Casi d’Uso

In UML, gli Use Case Diagram (UCD o diagrammi dei casi d’uso) sono diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Sono impiegati soprattutto nel contesto della Use Case View (vista dei casi d’uso) di un modello e in tal caso si possono considerare come uno strumento di rappresentazione dei requisiti funzionali di un sistema. Tuttavia, non è impossibile ipotizzare l’uso degli UCD in altri contesti; durante la progettazione, per esempio, potrebbero essere usati per modellare i servizi offerti da un determinato modulo o sottosistema ad altri moduli o sottosistemi. In molti modelli di processo software basati su UML, la Use Case View e gli Use Case Diagram che essa contiene rappresentano la vista più importante, attorno a cui si sviluppano tutte le altre attività del ciclo di vita del software (processi del genere prendono l’appellativo di processi Use Case Driven).

Diagramma dei Casi d’Uso del Terzo Ciclo

In questo diagramma vengono descritti tutti i casi d’uso presenti nel sistema.

Si può subito notare la presenza dei tre tipi di utenti precedentemente discussi. Si noti anche che vi è una catena di ereditarietà fra di essi: l’utente con privilegi di ROOT eredita dall’utente comune e quest’ultimo eredita dal-

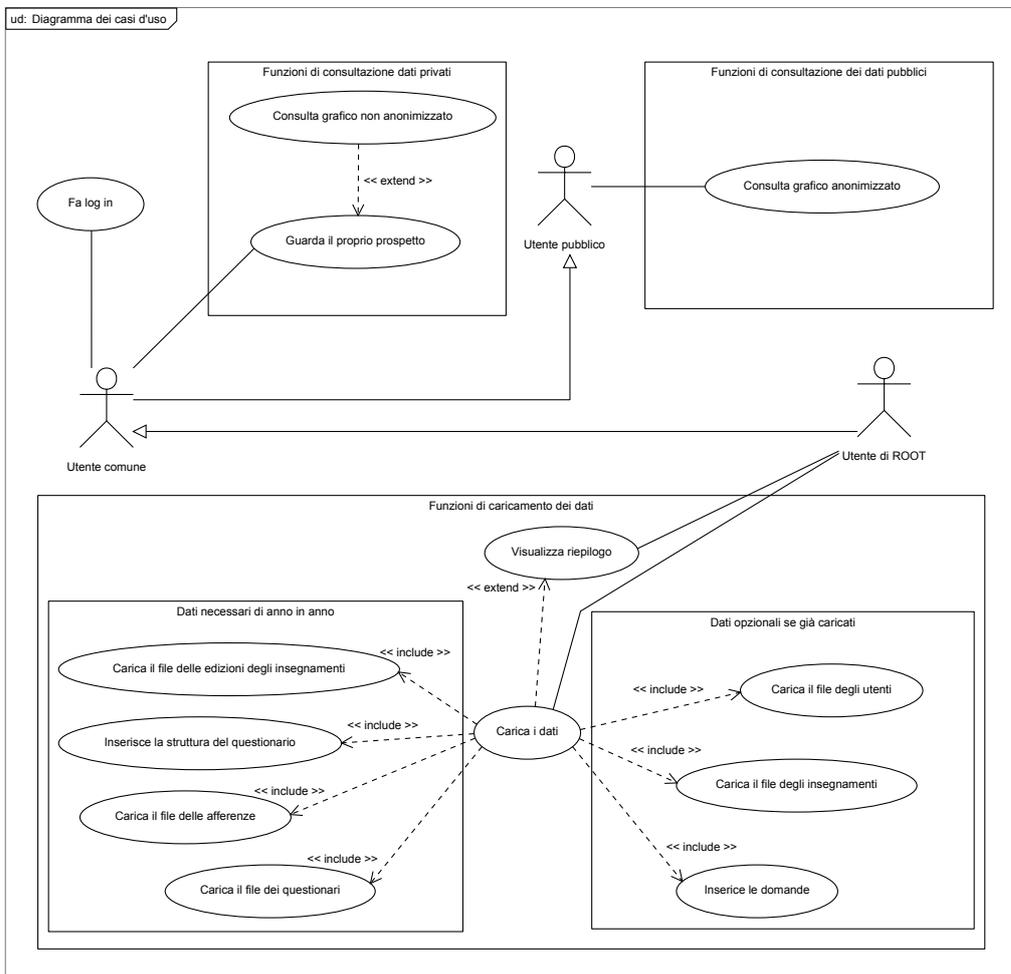


Figura 2.6: Il diagramma dei casi d'uso del terzo ciclo

l'utente pubblico. Questo avviene poiché, come spiegato nella **Tabella 2.1**, l'utente comune ha a disposizione, fra le altre, tutte le funzioni dell'utente pubblico e lo stesso vale per l'utente con privilegi di ROOT e per quello pubblico.

Andiamo ora a spiegare le funzioni alle quali ogni utente avrà accesso.

L'utente pubblico avrà accesso solo alle funzioni di consultazione dei dati pubblici che comprendono unicamente la visualizzazione del grafico anonimizzato.

L'utente comune potrà fare log in, dopodiché avrà accesso alle funzioni di consultazione dei dati privati che comprendono la visualizzazione del prospetto docente e quella del grafico non anonimizzato. Ovviamente ogni docente potrà consultare dati privati solo nel caso riguardino egli stesso.

L'utente con privilegi di ROOT avrà accesso alle funzioni di caricamento dei dati che comprendono la visualizzazione del riepilogo e il popolamento delle varie tabelle. Queste ultime si dividono in tabelle che dovranno essere popolate di anno in anno e tabelle il cui popolamento è opzionale se già effettuato. Questo perché se non vi sono nuovi utenti, insegnamenti o domande non si renderà necessario l'aggiornamento delle relative tabelle.

2.2.3 Diagramma di Attività

Descrizione dei Diagrammi di Attività

Il diagramma di attività definisce le attività da svolgere per realizzare una data funzionalità. Può essere utilizzato anche in fase di design per dettagliare un determinato algoritmo. In teoria ad ogni diagramma dei casi d'uso dovrebbe corrispondere un diagramma di attività. Più in dettaglio, un diagramma di attività definisce una serie di attività o flusso, anche in termini di relazioni tra di esse, chi è responsabile per la singola attività ed i punti di decisione.

Diagramma di Attività del Terzo Ciclo

Subito dopo la redazione del diagramma dei casi d'uso è stato redatto il diagramma di attività. Ciò ci permette di distillare tutte le attività che il programma dovrà eseguire per ottenere i risultati desiderati.

La prima attività che verrà effettuata dopo l'accesso è la visualizzazione della home page. Il visitatore a quel punto potrà visualizzare il grafico anonimizzato previa la scelta del CdS sul quale generarlo.

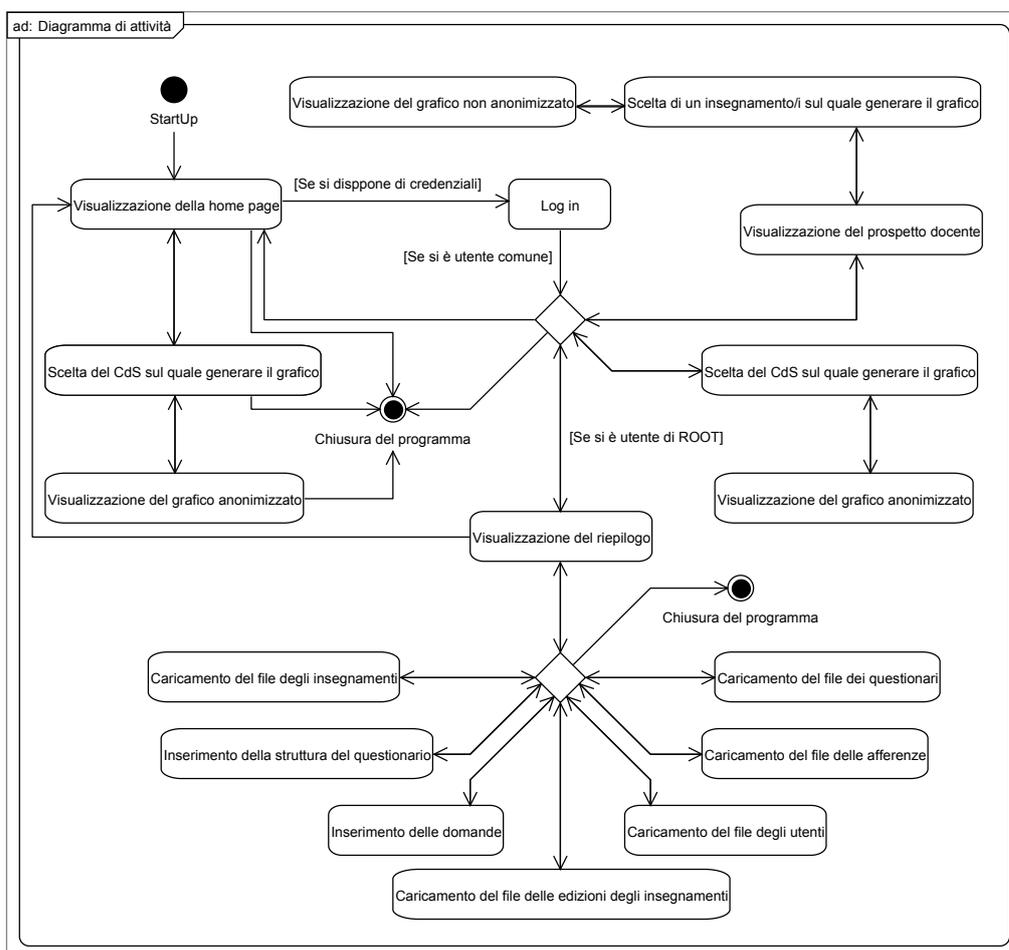


Figura 2.7: Il diagramma di attività del terzo ciclo

Se il visitatore dispone di credenziali potrà fare log in ed autenticarsi come utente con privilegi di ROOT o comune, avendo così accesso ad altre attività oltre quelle sopracitate.

Nel caso il visitatore sia un utente comune potrà visualizzare il proprio prospetto docente ed all'interno di esso scegliere su quali insegnamenti far generare il grafico non anonimizzato.

Se invece il visitatore è un utente con privilegi di ROOT, oltre a poter eseguire tutte le attività descritte fin ora, potrà visualizzare il riepilogo sullo stato delle tabelle e procedere alle attività di caricamento. Queste attività includono:

- il caricamento del file delle afferenze;
- il caricamento del file degli utenti;
- il caricamento del file degli insegnamenti;
- il caricamento del file delle edizioni degli insegnamenti;
- l'inserimento delle domande;
- l'inserimento della struttura del questionario;
- il caricamento del file dei questionari.

2.2.4 Diagramma di Stato

Descrizione dei Diagrammi di Stato

Un diagramma di stato è un tipo di diagramma utilizzato per descrivere il comportamento dei sistemi. I diagrammi di stato necessitano che il sistema descritto sia composto da un numero finito di stati. A volte questa condizione è verificata, altre volte invece si tratta di una ragionevole astrazione.

Diagramma di Stato del Terzo Ciclo

Il passo successivo alla redazione del diagramma di attività è quello di comprendere i vari stati che il programma dovrà assumere e documentarli nel diagramma di stato. Per fare ciò è risultato molto utile analizzare le attività che il programma dovrà eseguire, contenute nel precedente diagramma.

Sono stati identificati sei stati:

- home page;
- grafico anonimizzato;
- prospetto del docente;
- grafico non anonimizzato;
- riepilogo;

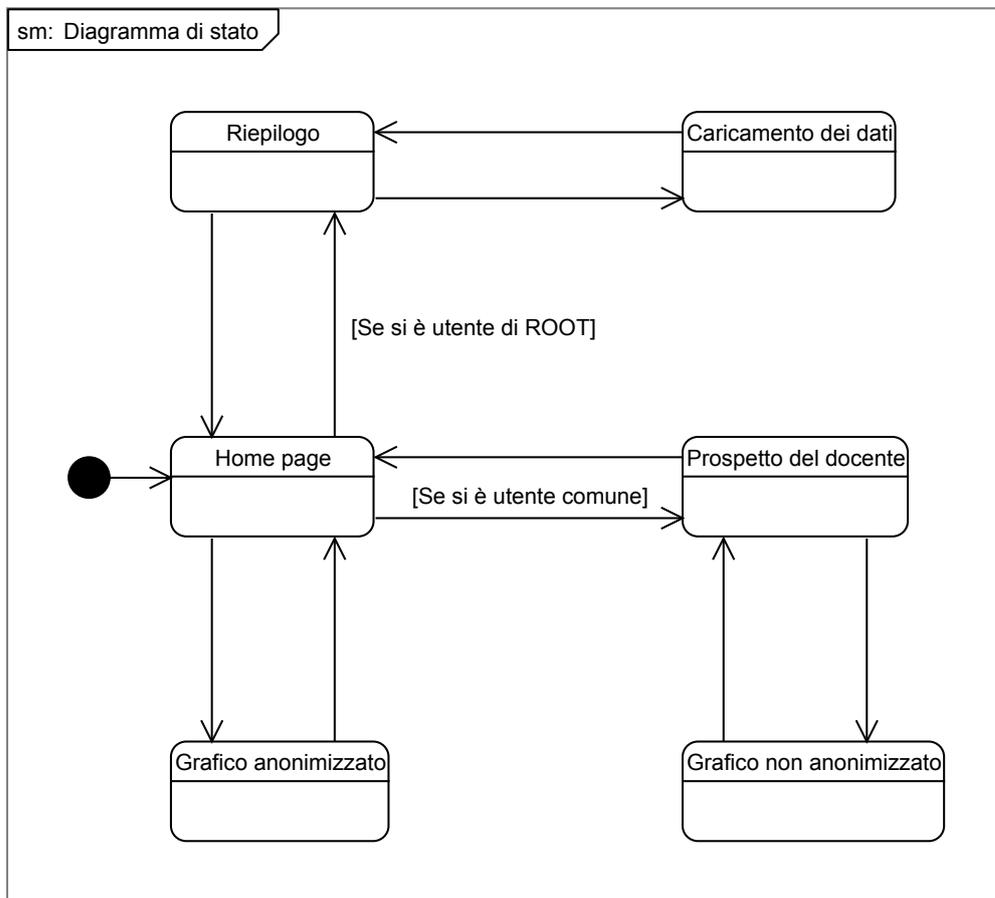


Figura 2.8: Il diagramma di stato del terzo ciclo

- caricamento dei dati.

Il primo stato che il programma assumerà è quello relativo alla home page, lo stato iniziale. A partire da questo stato il sistema potrà passare nei tre stati adiacenti, primo tra questi quello relativo al grafico anonimizzato.

In caso il visitatore sia un utente comune sarà possibile raggiungere dal primo stato anche quello relativo al prospetto docente; da quest'ultimo si potrà raggiungere quello relativo al grafico non anonimizzato.

Se poi il visitatore è un utente con privilegi di ROOT sarà possibile anche raggiungere lo stato relativo al riepilogo e da lì quello di caricamento dei dati.

2.2.5 Diagramma di Sequenza

Descrizione dei Diagrammi di Sequenza

I diagrammi di sequenza sono utilizzati per descrivere uno scenario. Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate; in pratica nel diagramma non compaiono scelte né flussi alternativi.

Normalmente da ogni diagramma di attività derivano uno o più diagrammi di sequenza; se per esempio il diagramma di attività descrive due flussi di azioni alternativi, se ne potrebbero ricavare due scenari, e quindi due diagrammi di sequenza alternativi. Dalla versione 2 di UML è stata introdotta la possibilità di indicare nello stesso diagramma anche delle sequenze alternative.

Il diagramma di sequenza descrive le relazioni che intercorrono, in termini di messaggi, tra Attori, Oggetti di business, Oggetti od entità del sistema che si sta rappresentando.

Diagrammi di Sequenza del Terzo Ciclo

È molto importante a questo punto poter capire l'ordine sequenziale con cui avverranno gli eventi nel programma. Questo è un aiuto fondamentale per comprendere a fondo, fin dal livello di progettazione, come il programma dovrà comportarsi.

Si è scelto di creare un diagramma di sequenza per ogni singolo utente allo scopo di avere diagrammi più chiari e leggibili.

Utente Pubblico L'unica cosa che l'utente pubblico potrà fare è quella di richiedere il grafico anonimizzato.

Il sottosistema GUI prenderà in carico la richiesta e chiederà a sua volta i dati per generare tal grafico al sottosistema di gestione del DB. Quest'ultimo una volta ricevuta tale richiesta invierà i dati in questione. Una volta ricevuti i dati necessari il sottosistema GUI genererà il grafico e lo mostrerà all'utente.

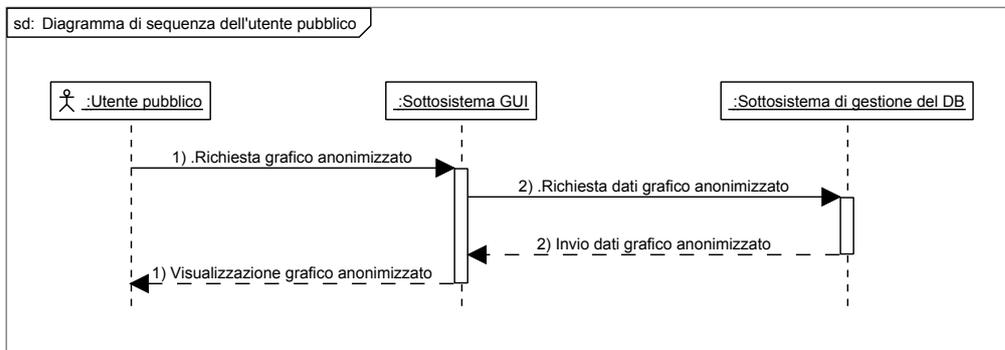


Figura 2.9: Il diagramma di sequenza del terzo ciclo relativo all'utente pubblico

Utente Comune Per dimostrare di essere tale, l'utente comune dovrà autenticarsi, quindi per prima cosa farà richiesta di log in.

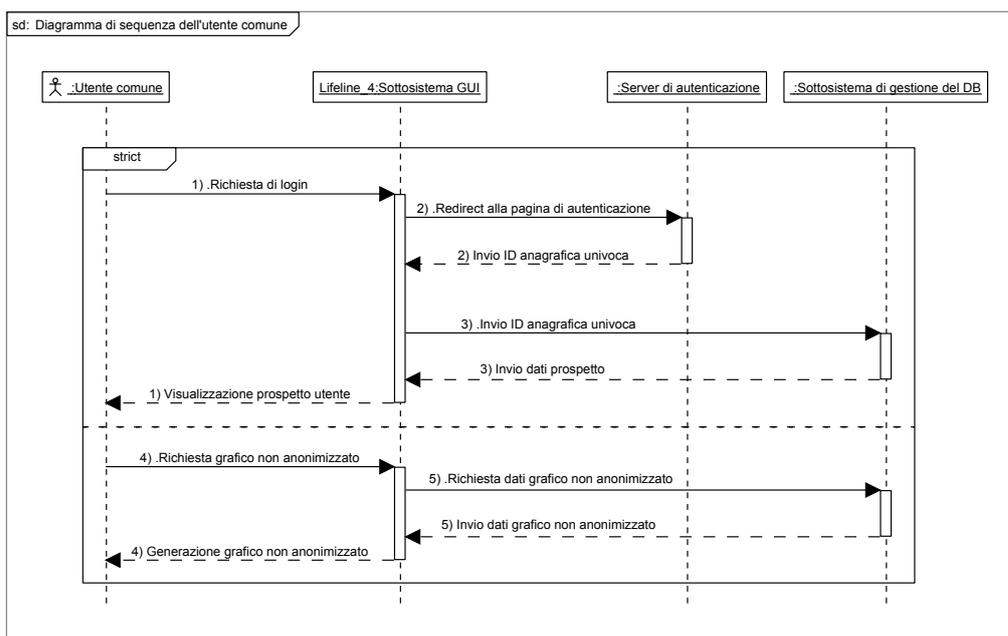


Figura 2.10: Il diagramma di sequenza del terzo ciclo relativo all'utente comune

Il sottosistema GUI reindirizzerà l'utente nella pagina di autenticazione dove quest'ultimo dovrà inserire le credenziali di ateneo. Se il log in va a buon fine il server di autenticazione restituisce al sottosistema GUI l'ID

anagrafica univoca dell'utente. Quest'ultima servirà a capire se il visitatore, anche se utente dell'università, risulta fra gli utenti del sistema; in tal caso il sottosistema di gestione del DB invierà a quello GUI i dati necessari alla generazione del prospetto docente.

Una volta visualizzato il prospetto docente l'utente potrà richiedere il grafico non anonimizzato. Come al solito questa richiesta verrà gestita dal sottosistema GUI che, una volta avuti i dati necessari dal sottosistema di gestione del DB, genererà il grafico e lo mostrerà all'utente.

Le operazioni relative all'autenticazione ed al grafico non anonimizzato sono situate nel fragment "strict" poiché è possibile eseguirle solo in sequenza: non si può accedere a dati non anonimizzati senza prima aver effettuato il log in.

Ovviamente, come previsto dalla catena di ereditarietà fra gli utenti, l'utente comune avrà accesso anche alle operazioni descritte nel diagramma di sequenza relativo all'utente pubblico.

Utente con privilegi di ROOT Oltre ad avere accesso alle funzioni che eredita dagli altri utenti, l'utente con privilegi di ROOT potrà visualizzare il riepilogo e popolare le tabelle, ovviamente dopo aver effettuato il log in.

Per prima cosa l'utente con privilegi di ROOT visualizzerà il riepilogo sullo stato delle tabelle. Per fare ciò egli farà richiesta di visualizzazione al sottosistema GUI che a sua volta richiederà i dati necessari al sottosistema di gestione del DB. Una volta ricevuti questi dati, il sottosistema GUI genererà il riepilogo e lo mostrerà all'utente.

Nel caso l'utente voglia popolare ex novo la base di dati, il primo file che dovrà caricare sarà quello delle afferenze.

Per fare ciò egli farà richiesta di caricamento al sottosistema GUI (eseguendo l'upload del relativo file) che lo farà processare dal sottosistema di acquisizione che ne estrarrà i dati tramite un parser. Una volta setacciati, i dati verranno inviati al sottosistema di gestione del DB che provvederà ad inserirli nella base di dati.

Questo procedimento è valido per tutti i popolamenti che avvengono tramite caricamento da file.

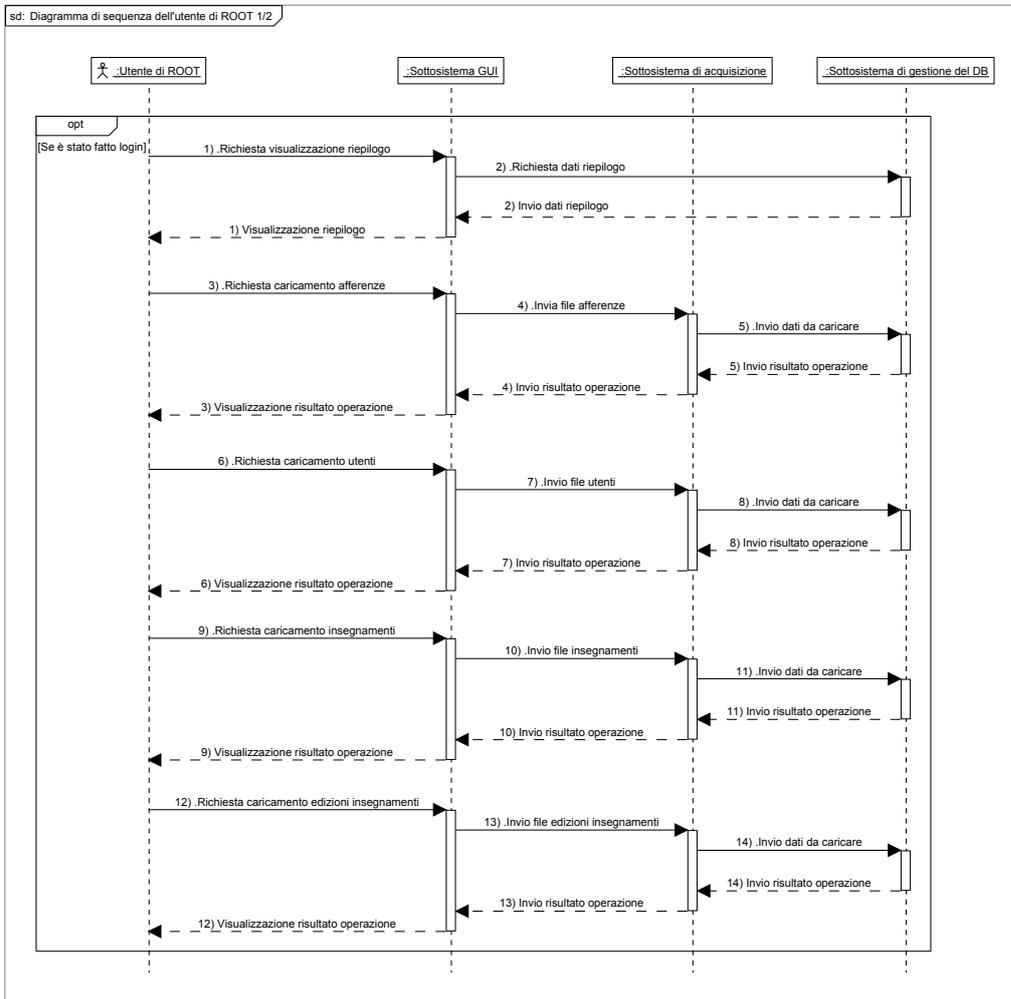


Figura 2.11: Prima metà del diagramma di sequenza del terzo ciclo relativo all'utente con privilegi di ROOT

Quando invece l'utente vorrà popolare le due tabelle i cui dati verranno inseriti con il popolamento manuale (DOMANDA e STRUTTURAQUESTIONARIO) dovrà innanzitutto richiedere al sottosistema GUI la pagina contenente i record già inseriti in suddette tabelle.

Questo sottosistema andrà a chiedere tali record al sottosistema di gestione del DB e una volta ricevuti i dati, andrà a generare la pagina che li mostra. In suddetta pagina l'utente potrà inserire nuove domande o nuove strutture del questionario.

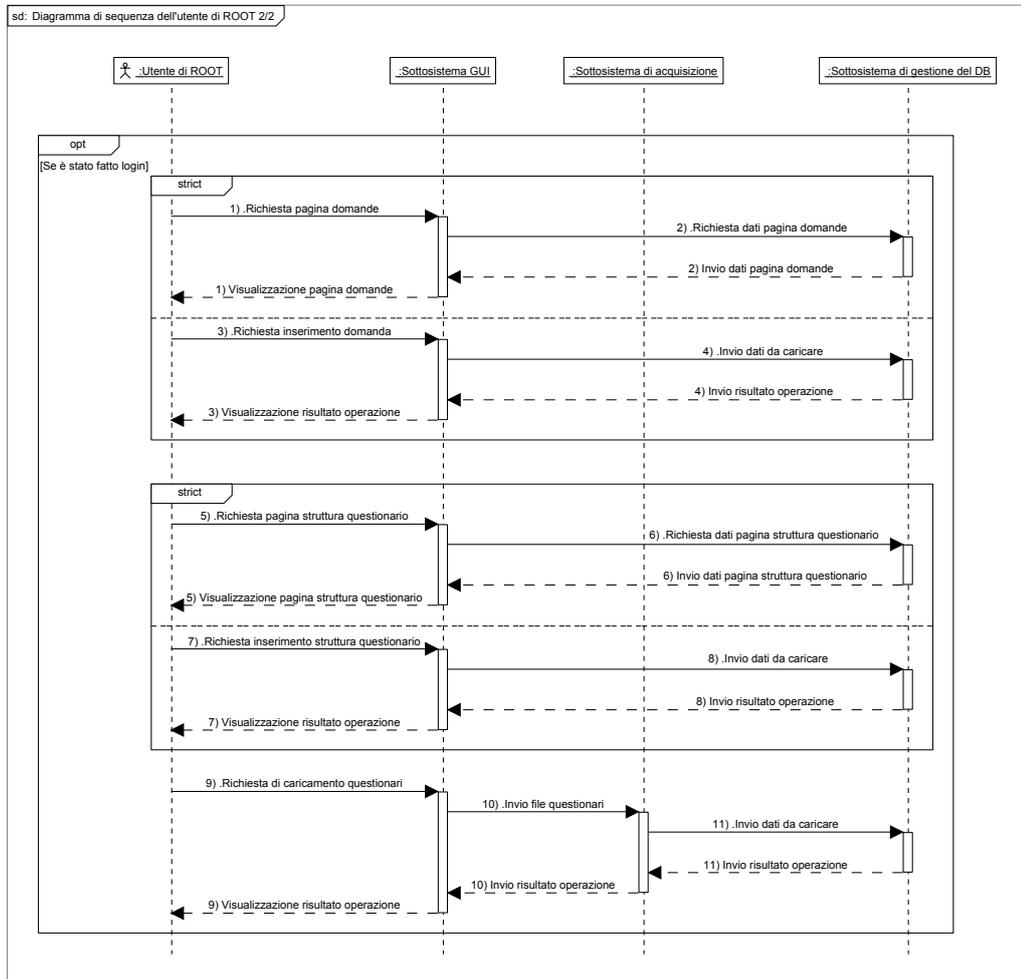


Figura 2.12: Seconda metà del diagramma di sequenza del terzo ciclo relativo all'utente con privilegi di ROOT

Ovviamente l'utente con privilegi di ROOT potrà, come per gli altri utenti, accedere ai servizi a lui riservati solo previa autenticazione; come si deduce dai fragment "OPT" nelle due metà del diagramma.

I fragment "STRICT" indicano invece che i messaggi racchiusi in essi devono essere processati nell'ordine stabilito.

2.2.6 Diagramma dei Componenti

Descrizione dei Diagrammi dei Componenti

Il diagramma dei componenti è un diagramma che ha lo scopo di rappresentare la struttura interna del sistema software modellato in termini dei suoi componenti principali e delle relazioni fra di essi. Per componente si intende una unità software dotata di una precisa identità, nonché responsabilità e interfacce ben definite.

Diagramma dei Componenti del Terzo Ciclo

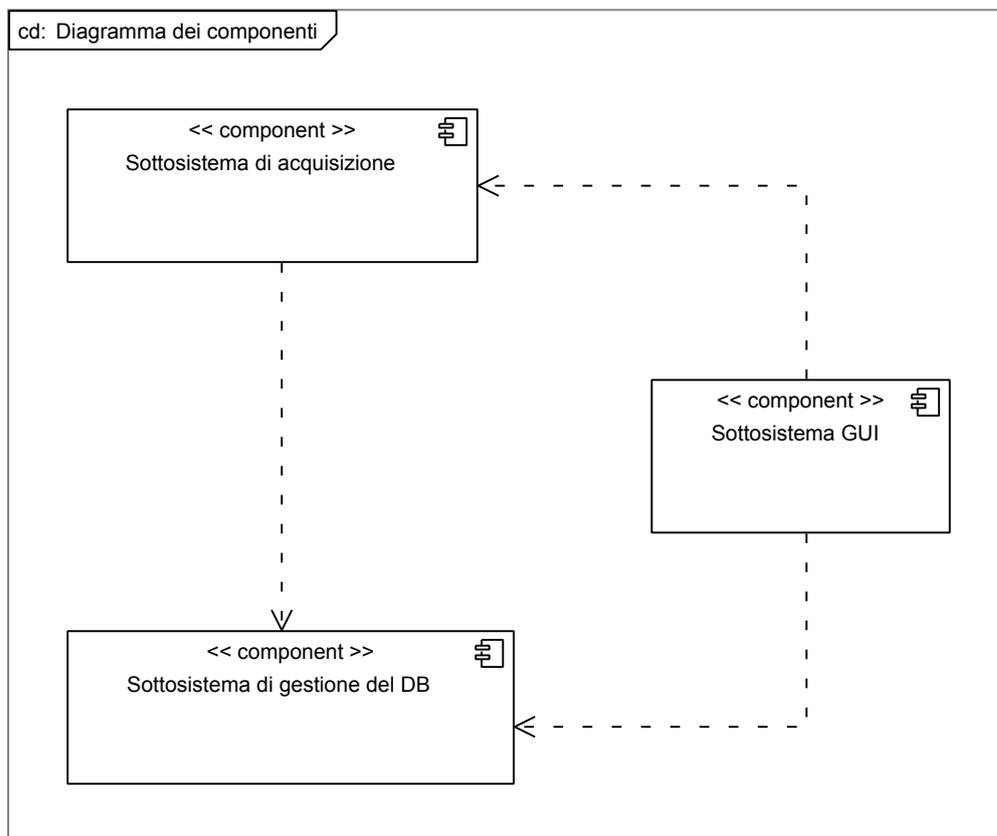


Figura 2.13: Il diagramma dei componenti del terzo ciclo

Sono stati identificati tre componenti principali a comporre il programma:

- il sottosistema GUI;
- il sottosistema di acquisizione;
- il sottosistema di Gestione del DB.

Ognuno di questi componenti è responsabile ed altamente specializzato nello svolgere i compiti a lui assegnati, compiti ampiamente descritti negli altri diagrammi.

Come il diagramma mostra, il componente che farà da gestore per tutto il sistema sarà il sottosistema GUI. Quest'ultimo dipende da due sottosistemi ovvero quello di gestione del DB e quello di acquisizione. Il sottosistema di acquisizione dipende anch'esso da quello di gestione del DB.

2.2.7 Diagramma delle Classi

Descrizione dei Diagrammi delle Classi

I diagrammi delle classi consentono di descrivere tipi di entità, con le loro caratteristiche e le eventuali relazioni fra questi tipi. Gli strumenti concettuali utilizzati sono il concetto di classe del paradigma object-oriented e altri correlati (per esempio la generalizzazione che è una relazione concettuale assimilabile al meccanismo object-oriented dell'ereditarietà).

Uno degli assunti fondamentali del paradigma a oggetti è che il concetto di classe e concetti correlati come l'ereditarietà o il polimorfismo, si prestino a rappresentare in modo diretto e intuitivo la realtà in qualsiasi ambito (per usare le parole di Grady Booch, “un oggetto è qualsiasi cosa a cui si possa pensare”). I diagrammi delle classi UML sono basati su versioni astratte di tali concetti e possono essere utilizzati per descrivere sostanzialmente qualsiasi contesto a qualsiasi livello di astrazione (enfaticandone però, solo alcuni aspetti). Di conseguenza, UML prevede un loro impiego a livello di analisi e in particolare analisi del dominio (ovvero la descrizione del contesto in cui un sistema software deve operare), ma anche a livello di progettazione (nella descrizione della struttura interna del sistema, dei suoi componenti e delle loro relazioni).

Diagramma delle Classi del Terzo Ciclo

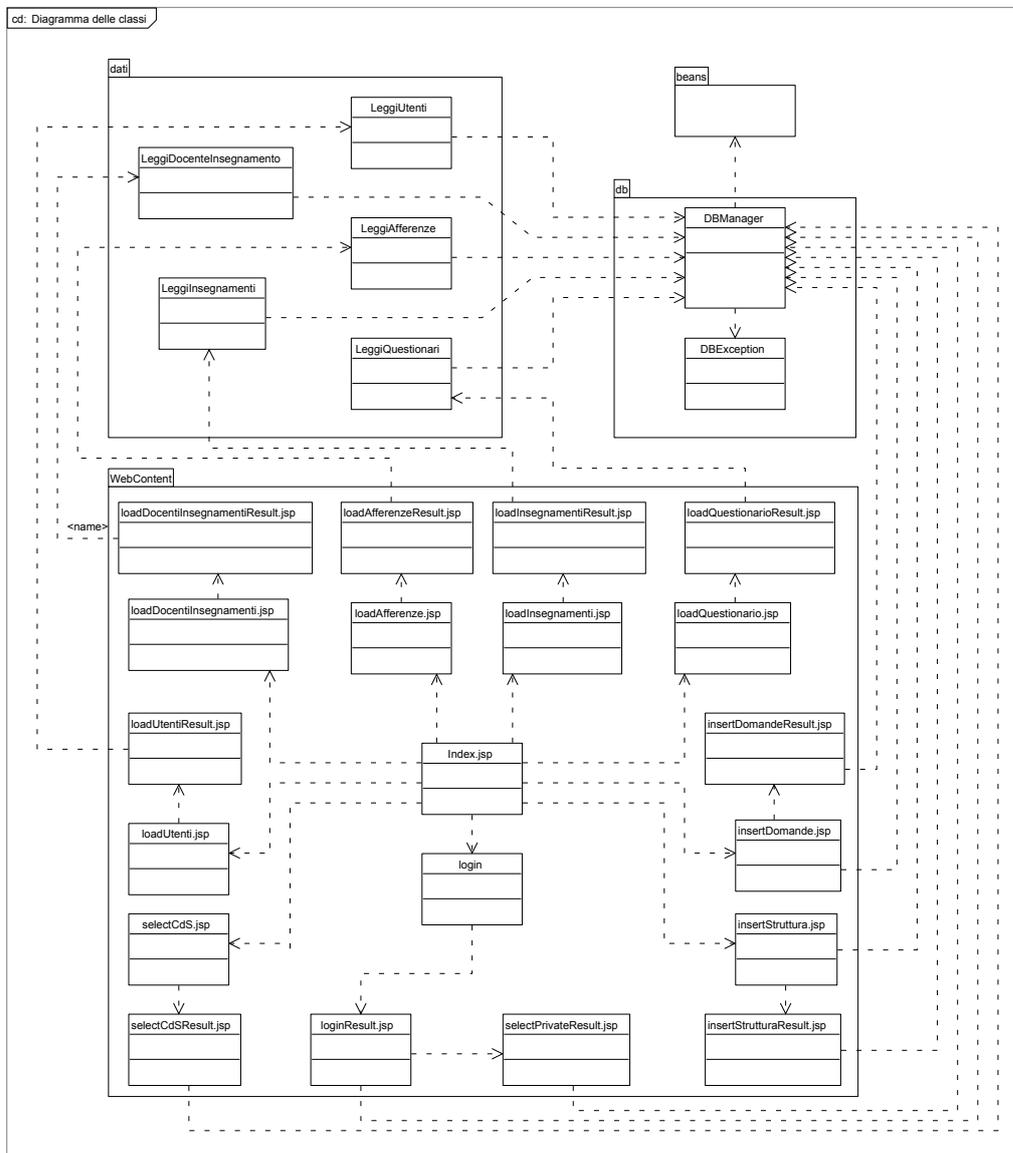


Figura 2.14: Il diagramma delle classi del terzo ciclo

Come si può subito notare il diagramma delle classi è suddiviso in package.

Il package denominato “dati” sarà quello che conterrà le classi afferenti al sottosistema di acquisizione; quelle classi quindi che conterranno i vari file:

- `LeggiUtenti`;

- LeggiDocenteInsegnamento;
- LeggiAfferenze;
- LeggiInsegnamenti;
- LeggiQuestionari.

Il package denominato “WebContent” è quello che contiene tutte le pagine jsp, quindi all’atto pratico tale package contiene il sottosistema GUI.

Il package denominato “db” è quello che contiene le classi preposte alla gestione della base di dati.

L’ultimo package rimasto è quello denominato “beans”: esso contiene tutti i bean necessari al funzionamento di Hibernate.

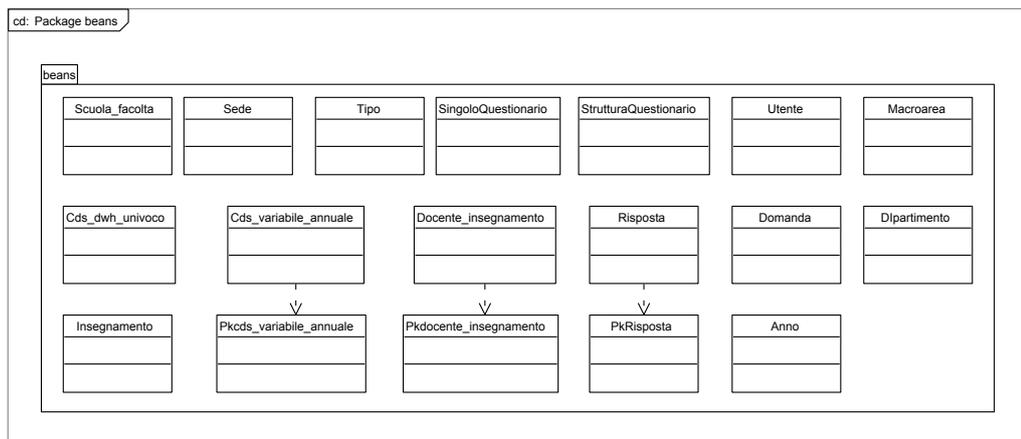


Figura 2.15: Il package denominato beans

Ogni classe in questo package rappresenta una tabella, eccezion fatta per quella che rappresentano delle primary key composite:

Pkcds_variabale_annuale è il bean che verrà usato dalla classe Cds_variabale_annuale come primary key;

Pkdocente_insegnamento è il bean che verrà usato dalla classe Docente_insegnamento come primary key;

PkRisposta è il bean che verrà usato dalla classe Risposta come primary key.

La prima entità ad essere invocata sarà sicuramente la pagina index.jsp. Questa pagina richiamerà tutte le pagine per la richiesta di funzioni che a loro volta richiameranno quelle, che attraverso l'ausilio degli altri sottosistemi, forniranno tali funzioni.

2.2.8 Diagramma di Dispiegamento

Descrizione dei Diagrammi di Dispiegamento

Il diagramma di dispiegamento è un diagramma di tipo statico usato per descrivere un sistema in termini di risorse hardware, dette nodi, e di relazioni fra di esse. Spesso si utilizza un diagramma che mostra come le componenti software sono distribuite rispetto alle risorse hardware disponibili sul sistema.

Diagramma di Dispiegamento del Terzo Ciclo

Come possiamo vedere nel diagramma, vi sarà il nostro server in cui gireranno Oracle 11g e Apache Tomcat. Quest'ultimo eseguirà l'applicativo SERED, che si interfacerà sia con il DB all'interno di Oracle sia con Shibboleth all'interno del server di autenticazione.

Il diagramma mostra anche come la macchina dell'utente si collegherà all'applicativo tramite il web.

2.3 Scelte Progettuali

In questa sezione verranno spiegate le motivazioni delle scelte progettuali prese.

Senza altro la prima decisione presa è stata quella di adottare lo stack tecnologico discusso nei requisiti (vedi **Sezione 2.1**). L'adozione di tale stack è stata una scelta obbligata poiché altrimenti l'applicativo non si sarebbe potuto integrare con i sistemi universitari sui server di ateneo.

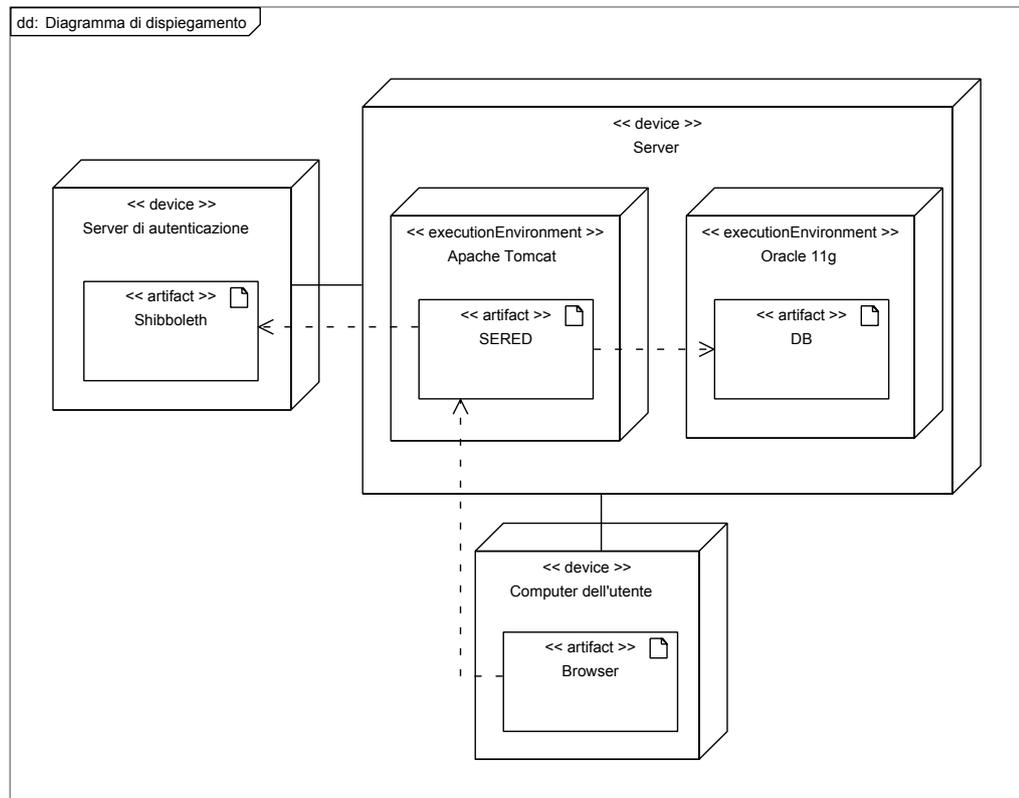


Figura 2.16: Il diagramma di dispiegamento del terzo ciclo

Un'altra importante scelta da motivare è quella di offrire solo un numero limitato di funzioni di consultazione dei dati. Il motivo di questa decisione risiede nell'idea che il core value di questo lavoro di tesi consista nell'unificare ed uniformare dati provenienti da fonti così diverse dando loro una coerenza interna. Ad ogni modo la tecnica di sviluppo prescelta (vedi **Sezione 1.2**) permette, attraverso l'aggiunta di altre iterazioni, di aggiungere qualsivoglia funzione.

2.4 Possibili Sviluppi Evolutivi

Tra le possibili modifiche evolutive, la più importante è sicuramente quella riguardante l'infittimento dei livelli di utenza: sarebbe interessante aggiungere delle tipologie di utenti che permettano, non solo ai docenti di consultare

i dati riguardanti loro, ma ad altre figure di rilievo di accedere ai dati di propria competenza.

Ad esempio:

il coordinatore di un CdS dovrebbe avere accesso a tutti i dati riguardanti quest'ultimo;

il direttore di un Dpt dovrebbe avere accesso a tutti i dati riguardanti i CdS afferenti al dipartimento che egli dirige;

il preside di una Scl dovrebbe avere accesso a tutti i dati riguardanti i CdS afferenti alla scuola che egli presiede;

rettore e prorettori dovrebbero avere accesso alla totalità dei dati.

Parte II

Illustrazione del Sistema

Capitolo 3

L'Utente Pubblico

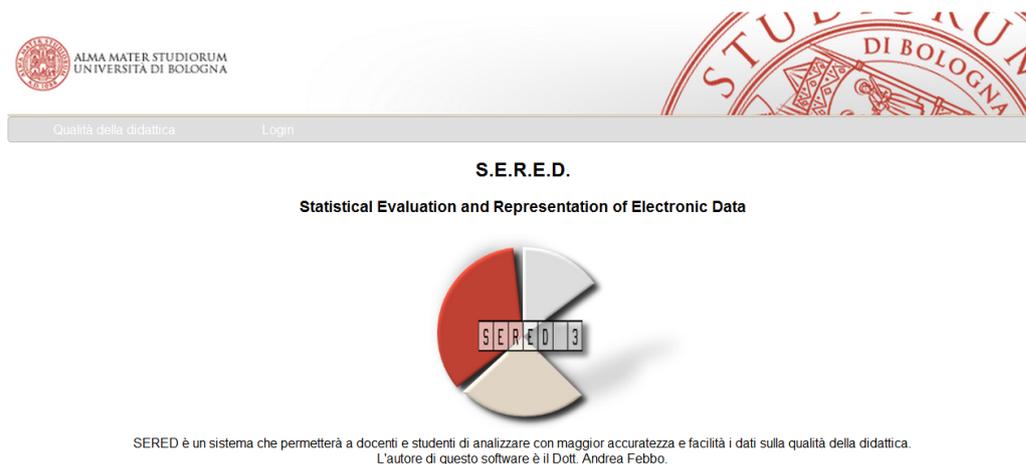


Figura 3.1: La home page del sistema vista dall'utente pubblico

In questo capitolo andremo a spiegare l'uso delle funzioni riservate all'utente pubblico.

Come detto in precedenza l'utente pubblico è chiunque sia interessato alla qualità della didattica dell'ateneo bolognese, studenti e non: per questo motivo non ci sarà bisogno di nessuna autenticazione.

Per ora il sistema offre solo una funzione di consultazione dei dati all'utente pubblico: la visualizzazione del grafico anonimizzato. Le motivazioni di questa scelta sono illustrate nella **Sezione 2.3**.

3.1 Il Grafico Anonimizzato

Questa funzione consiste nella visualizzazione di istogrammi raffiguranti la soddisfazione degli studenti sui vari CdS. Per accedere a questa funzione l'utente dovrà cliccare sul tasto "Qualità della didattica" posto nel menu (Figura 3.1).

Seleziona il Corso di Studio

Anno: 2010 / 2011 ▾

Scuola	Corso di Studio
<input type="radio"/> Agraria	<input type="radio"/> CONSULENTE DEL LAVORO E DELLE RELAZIONI AZIENDALI - L - Bologna
<input type="radio"/> Architettura	<input type="radio"/> GIURISPRUDENZA - LS - Bologna
<input type="radio"/> Chimica Industriale	<input type="radio"/> GIURISPRUDENZA - LMCU - Ravenna
<input type="radio"/> Conservazione dei beni culturali	<input checked="" type="radio"/> GIURISPRUDENZA - LMCU - Bologna
<input type="radio"/> Economia Bologna	<input type="radio"/> GIURISTA D'IMPRESA E DELLE AMMINISTRAZIONI PUBBLICHE - L - Ravenna
<input type="radio"/> Economia Forlì	<input type="radio"/> OPERATORE GIURIDICO ITALO-FRANCESE - L - Bologna
<input type="radio"/> Economia Rimini	<input type="radio"/> OPERATORE INFORMATICO-GIURIDICO - L - Bologna
<input type="radio"/> Farmacia	<input type="radio"/> SCIENZE GIURIDICHE - L509 - Bologna
<input checked="" type="radio"/> Giurisprudenza	
<input type="radio"/> Ingegneria	
<input type="radio"/> Lettere e Filosofia	
<input type="radio"/> Lingue e letterature straniere	
<input type="radio"/> Medicina e Chirurgia	
<input type="radio"/> Medicina Veterinaria	
<input type="radio"/> Psicologia	
<input type="radio"/> Scienze della formazione	
<input type="radio"/> Scienze MM FF NN	
<input type="radio"/> Scienze Motorie	
<input type="radio"/> Scienze politiche Bologna	
<input type="radio"/> Scienze politiche Forlì	
<input type="radio"/> Scienze Statistiche	
<input type="radio"/> Seconda Facoltà di Ingegneria	
<input type="radio"/> SSLIMIT	

Figura 3.2: Il menù di scelta del CdS

A questo punto l'utente dovrà scegliere innanzitutto l'anno e la scuola per il quale è interessato a visualizzare i grafici. Una volta scelti questi due parametri apparirà un ulteriore menù contenente tutti i CdS afferenti in quell'anno a quella scuola; l'utente potrà così scegliere di quale corso vorrà visualizzare i dati.

Una volta scelto il CdS si aprirà una pagina contenente gli istogrammi di soddisfazione. Come specificato varie volte in precedenza, l'utente pubblico potrà visualizzare dati solo in forma anonima.

La pagina contenente tali istogrammi ne avrà uno per ogni domanda contenuta nel questionario dell'anno selezionato. Le colonne di questi istogrammi rappresentano gli insegnamenti afferenti al CdS scelto e la loro altezza rappresenta la percentuale di soddisfazione degli studenti su quella domanda per quell'insegnamento. Le suddette colonne sono ordinate in maniera discen-

Grafico della soddisfazione

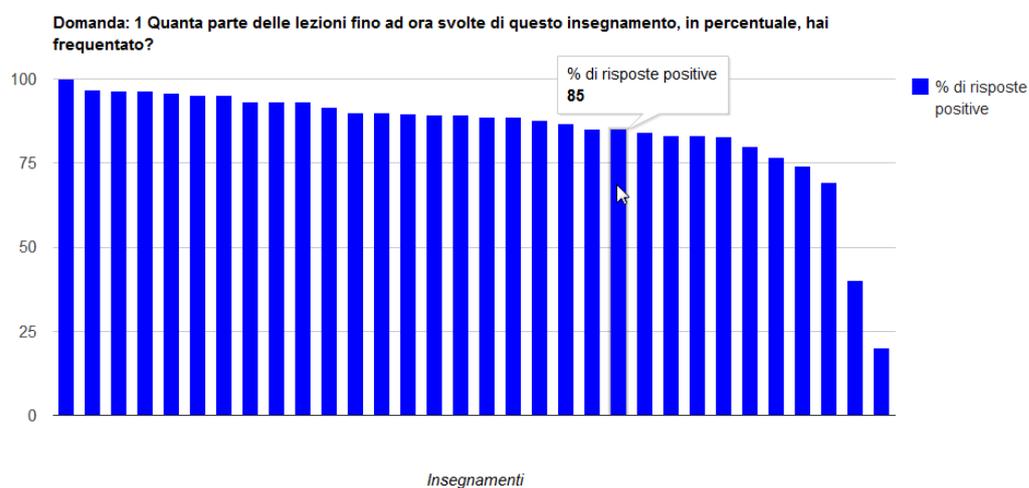


Figura 3.3: Esempio di istogramma di soddisfazione anonimizzato

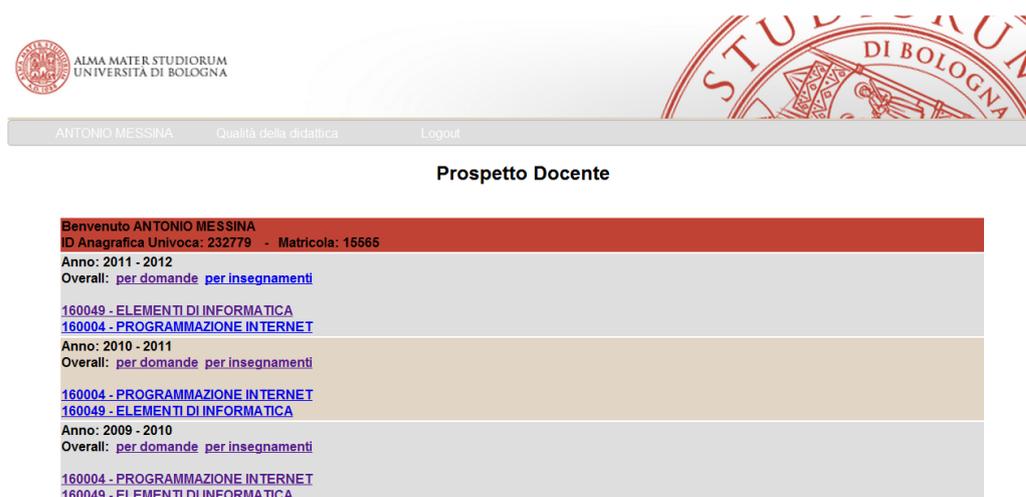
dente: questo sia per facilitare l'interpretazione dei dati sia per massimizzare l'anonimato degli insegnamenti e quindi dei professori.

Passando il mouse sulle colonne apparirà un tooltip contenente la percentuale di risposte positive. Questo vale per tutti i grafici generati dal sistema.

Capitolo 4

L'Utente Comune

Se il visitatore possiede i privilegi di utente comune, ad esempio un docente, non dovrà fare altro che autenticarsi usando il single sign-on di ateneo. Per raggiungere la pagina di autenticazione si deve cliccare sul tasto “Login” nel menù.



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ANTONIO MESSINA Qualità della didattica Logout

Prospetto Docente

Benvenuto ANTONIO MESSINA
ID Anagrafica Univoca: 232779 - Matricola: 15565

Anno: 2011 - 2012
Overall: [per domande](#) [per insegnamenti](#)

[160049 - ELEMENTI DI INFORMATICA](#)
[160004 - PROGRAMMAZIONE INTERNET](#)

Anno: 2010 - 2011
Overall: [per domande](#) [per insegnamenti](#)

[160004 - PROGRAMMAZIONE INTERNET](#)
[160049 - ELEMENTI DI INFORMATICA](#)

Anno: 2009 - 2010
Overall: [per domande](#) [per insegnamenti](#)

[160004 - PROGRAMMAZIONE INTERNET](#)
[160049 - ELEMENTI DI INFORMATICA](#)

Figura 4.1: Il prospetto docente

Una volta autenticatisi come utente comune il visitatore visualizzerà il proprio prospetto docente (vedi **Figura 4.1**). Questo prospetto contiene una tabella suddivisa per anni con tutti gli insegnamenti tenuti dall'utente.

4.1 Il Grafico non Anonimizzato

È possibile cliccare sui singoli insegnamenti o sugli overall per visualizzare i grafici di soddisfazione.

4.1.1 Overall per Domande

Se si sceglie la voce “per domande” si visualizzerà una pagina contenente vari istogrammi rappresentanti, come nel grafico anonimizzato, le domande; le barre rappresentano invece la percentuale di soddisfazione sugli insegnamenti.

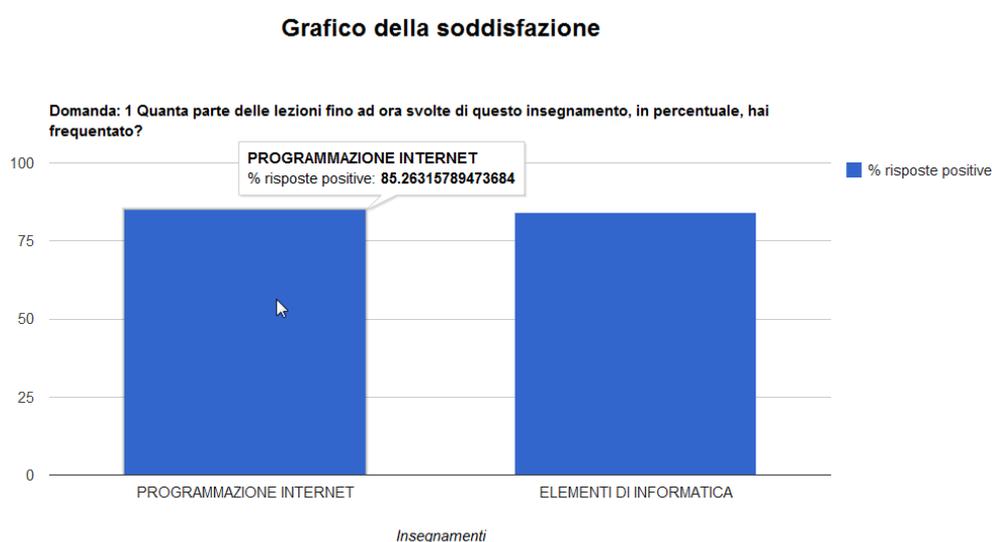


Figura 4.2: Esempio di istogramma di soddisfazione non anonimizzato (per domande)

La differenza fondamentale è che i dati non sono più anonimizzati (vedi **Figura 4.2**), cosicché il docente possa capire ove è necessario lavorare per migliorare la qualità della didattica.

4.1.2 Overall per Insegnamenti

Se invece si sceglie la voce “per insegnamenti” la pagina che verrà visualizzata conterrà tanti grafici quanti sono gli insegnamenti e le barre rappresenteranno la percentuale di soddisfazione sulle domande (vedi **Figura 4.3**).

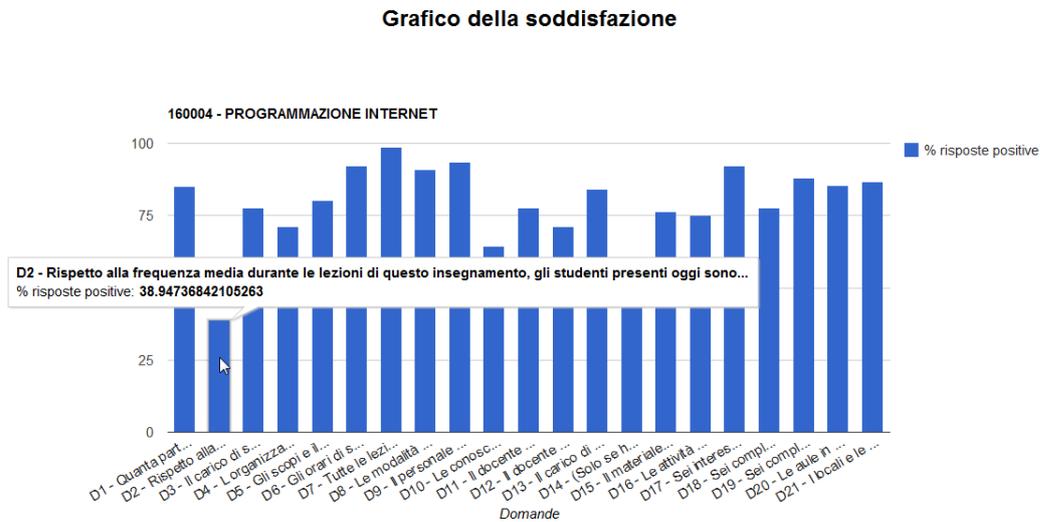


Figura 4.3: Esempio di istogramma di soddisfazione non anonimizzato (per insegnamenti)

Si è deciso di dare la possibilità all'utente di scegliere due modi di visualizzare i dati (per domande e per insegnamenti) così da massimizzare l'interpretabilità di questi ultimi.

4.1.3 Singoli Insegnamenti

L'utente potrà scegliere di visualizzare il grafico su un singolo insegnamento. Le barre dell'istogramma che verrà generato rappresenteranno, come nel caso dell'overall per insegnamenti, la percentuale di soddisfazione sulle domande.

Capitolo 5

L'Utente con Privilegi di ROOT

Se l'utente possiede privilegi di ROOT avrà accesso, oltre alle funzioni disponibili agli altri utenti, al caricamento dei dati.

Il caricamento dei dati prevede che si popolino una serie di tabelle attraverso dei file CSV o tramite inserimento manuale.

The screenshot shows the Alma Mater Studiorum University of Bologna interface. At the top, there is a navigation bar with the user's name 'ANDREA FEBBO' and several menu items: 'Riepilogo', 'Caricamento dati', 'Qualità della didattica', and 'Logout'. Below the navigation bar, the title 'Riepilogo' is centered. The main content area is divided into two columns. The left column, titled 'Obbligatori per l'anno: 2010 / 2011', lists four items, each with a green checkmark and a 'Carica' link: 'Afferenze (232)', 'Edizioni Insegnamenti (12718)', 'Struttura Questionario (21)', and 'Questionari (193736)'. The right column, titled 'Opzionali se già caricati', lists three items, each with a green checkmark, a timestamp, and a 'Carica' link: 'Utenti (10845)' (ultimo caricamento 29/05/2012 - 22:41:47), 'Insegnamenti (12813)' (ultimo caricamento 07/06/2012 - 21:58:02), and 'Domande (26)' (ultimo caricamento 20/05/2012 - 17:00:57).

Figura 5.1: Il riepilogo

Per fare il punto della situazione e ricordarsi quali passi sono stati compiuti e quali no nel caricamento dei dati è stato ideato un riepilogo. Tale riepilogo (vedi **Figura 5.1**) mostra quali passi sono stati effettuati, attraverso un segno di spunta verde, e quali no, attraverso una ics rossa.

Come evidenziato nel diagramma dei casi d'uso (vedi **Sottosezione 2.2.2**), vi sono due insiemi di caricamenti presenti nel riepilogo: obbligatori per l'an-

no corrente e opzionali se già caricati. Andiamo ora a descriverli più nello specifico.

Obbligatorî per l'anno corrente: questi caricamenti sono quelli che dovranno essere obbligatoriamente effettuati di anno in anno. Attraverso di essi si “fotograferà” l'università, le edizioni degli insegnamenti e i questionari: passaggi necessari tenendo conto della variabilità di questi dati. I caricamenti che appartengono a questo insieme sono:

- le afferenze;
- le edizioni degli Insegnamenti;
- la struttura del questionario;
- i questionari.

Opzionali se già caricati: questi caricamenti, teoricamente, potrebbero essere effettuati anche una sola volta poiché potrebbero non esserci mai nuovi utenti, insegnamenti o domande. I caricamenti che appartengono a questo insieme sono:

- gli utenti;
- gli insegnamenti;
- le domande.

Il formato dei file che si andrà a caricare è stato deciso in collaborazione con gli stakeholder; non è detto che tutte le informazioni presenti in essi verranno usate dal programma, alcune potrebbero esistere solo per la comodità di chi li ha creati.

Andiamo ora a descrivere le singole funzionalità di caricamento una ad una nell'ordine nel quale è auspicabile eseguirle in uno scenario di popolazione ex novo del DB.

5.1 Afferenze

Come detto anche in precedenza (vedi **Sottosezione 2.1.1**) il caricamento del file delle afferenze “fotografa” la struttura dell’ateneo in un determinato anno; questo è sicuramente il primo passo da compiere per popolare il DB.

Questa funzione popola la tabella ANNO, se il record che si riferisce all’anno per il quale si sta caricando non è già presente, e la tabella CDS_VARIABILE_ANNUALE.

Il formato del file delle afferenze deve essere come quello mostrato nella **Tabella 5.1**.

CdS	IDCSV	IDDWH	AA	IDSCL	ScI	Dpt	MA
Storia	7	843	2010/11	12	Lettere	0	0
Moda	25	756	2010/11	12	Lettere	0	0
Economia	14	805	2010/11	54	Economia	0	0

Tabella 5.1: Il formato del file delle afferenze

In questo file il dato riguardante l’anno accademico viene preso dalla relativa colonna; in tutti gli altri CSV da qui in poi questo dato dovrà essere inserito nel nome file stesso.

5.2 Utenti

Caricando il file degli utenti si popolerà soltanto la tabella UTENTE. Il formato di questo file deve essere come quello mostrato nella **Tabella 5.2**.

IDANAGRAFICAUNIVOCA	MATRICOLA	COGNOME	NOME
12345	67890	Mario	Rossi
09876	54321	Luigi	Bianchi
13579	24680	Anna	Neri

Tabella 5.2: Il formato del file degli utenti

Come si può dedurre dal formato del file il campo denominato TIPO presente nella tabella non verrà popolato attraverso caricamento. Il valore di default di questo campo è 'C' (vedi **Appendice A.14**), che sta per utente

comune. Il primo record di questa tabella deve essere quindi creato da script, creando così il primo utente con privilegi di ROOT (con il tipo settato ad 'R').

Una volta creato il primo utente con privilegi di ROOT si potrà procedere con i caricamenti.

Ogni volta che bisognerà aggiungere nuovi utenti con privilegi di ROOT lo si dovrà fare da script: creando un nuovo utente con il tipo settato ad 'R', oppure attraverso l'update di un utente già esistente.

5.3 Insegnamenti

Caricando il file degli insegnamenti si popolerà soltanto la tabella INSEGNAMENTO. Il formato di questo file deve essere come quello mostrato nella **Tabella 5.3**.

IDINSEGNAMENTO	NOMEINSEGNAMENTO
110004	Diritto del lavoro
110005	Diritto commerciale
110006	Diritto amministrativo

Tabella 5.3: Il formato del file degli insegnamenti

5.4 Edizioni Insegnamenti

Caricando il file delle edizioni degli insegnamenti si popolerà soltanto la tabella DOCENTE_INSEGNAMENTO. Il formato di questo file deve essere come quello mostrato nella **Tabella 5.4**.

È da notare che, come mostrato nella tabella, un solo professore può tenere più insegnamenti in un solo anno.

Per cause di forza maggiore possiamo disporre delle informazioni contenute in questo file solo con il campo MATRICOLA come riferimento ai docenti poiché le fonti di suddette informazioni non usano l'ID anagrafica univoca. Questo campo solitamente è univoco, ma tale univocità non è garantita. Il

MATRICOLA	IDINSEGNAMENTO
12345	110004
12345	110005
54321	110006

Tabella 5.4: Il formato del file delle edizioni degli insegnamenti

software quindi controllerà, per ogni matricola presente nel file, se vi è associato un solo IDANAGRAFICAUNIVOCA, in caso contrario scarcerà la riga (vedi **Sezione 5.8**).

5.5 Domande

Le domande fanno parte di quell'insieme di dati che verranno inseriti manualmente. A tale scopo è stata creata una pagina che permetterà l'inserimento e la cancellazione delle domande (vedi **Figura 5.2**).

Inserisci Domande

Id	Domanda	Cancella
9	Il carico di studio complessivo degli insegnamenti ufficialmente previsti in questo anno accademico è accettabile?	
10	L'organizzazione complessiva (orario, esami intermedi e finali) degli insegnamenti ufficialmente previsti in questo anno accademico è accettabile?	
11	Gli scopi e il programma di questo insegnamento sono stati chiari sin dall'inizio?	
12	I nessi fra le diverse parti dell'insegnamento sono chiari?	
13	Gli orari di svolgimento dell'attività didattica sono rispettati?	
14	Tutte le lezioni sono state svolte o comunque presiedute dal titolare dell'insegnamento?	
15	Le modalità d'esame sono state definite in modo chiaro?	
16	Il personale docente è effettivamente reperibile per chiarimenti e spiegazioni?	
17	Se sono presenti aree di sovrapposizione tra gli argomenti trattati in questo insegnamento e quelli di altri insegnamenti, queste sono utili? (se non sono presenti tali sovrapposizioni annerite il tondino "non presenti")	
18	Le conoscenze preliminari da te possedute sono risultate sufficienti per la comprensione degli argomenti trattati?	

Domanda:

Figura 5.2: La pagina di gestione delle domande

Per inserire una nuova domanda sarà sufficiente scriverla nell'apposito spazio e cliccare sul tasto aggiorna.

Per cancellare le domande, invece, basterà cliccare sull'icona del bidoncino posto alla destra di ognuna di esse. Tale cancellazione sarà possibile solo se la domanda che si sta tentando di eliminare non è stata già inserita nella struttura dei questionari di un anno.

Questa funzione popolerà solo la tabella DOMANDA.

5.6 Struttura Questionario

Anche le strutture dei questionari, come le domande, fanno parte di quell'insieme di dati che verranno inseriti manualmente ed anche per esse è stata creata una pagina per permetterne la gestione (vedi **Figura 5.3**).

Struttura Questionario

Anno:

Posizione	Scala	Domanda	Id
<input type="text" value="3"/>	<input type="text" value="4"/>	Il materiale didattico e i testi indicati per seguire l'insegnamento e preparare l'esame sono di facile reperibilità?	25
<input type="text" value="4"/>	<input type="text" value="4"/>	Le attività didattiche integrative (esercitazioni, laboratori, seminari, ecc.) sono utili ai fini dell'apprendimento? (se non sono previste tali attività annerite il tondino "non previste")	26
<input type="text" value="5"/>	<input type="text" value="4"/>	Le aule in cui si svolgono le lezioni sono adeguate (si vede, si sente, si trova posto)?	27
<input type="text" value="6"/>	<input type="text" value="4"/>	I locali e le attrezzature per le attività didattiche integrative (esercitazioni, laboratori, seminari, ecc.) sono adeguati? (se non sono previste tali attività annerite il tondino "non previste")	28
<input type="text" value="7"/>	<input type="text" value="4"/>	Sei interessato/a agli argomenti di questo insegnamento (indipendentemente da come è stato svolto)?	29
<input type="text" value="8"/>	<input type="text" value="4"/>	Sei complessivamente soddisfatto/a di come è stato svolto questo insegnamento?	30
<input type="text" value="9"/>	<input type="text" value="4"/>	Sei complessivamente soddisfatto/a del corso di laurea al quale sei iscritto/a?	31
<input type="text" value="10"/>	<input type="text" value="4"/>	Sei complessivamente soddisfatto/a della facoltà alla quale sei iscritto/a?	32
<input type="text" value="1"/>	<input type="text" value="10"/>	Quanta parte delle lezioni fino ad ora svolte di questo insegnamento, in percentuale, hai frequentato?	35
<input type="text" value="2"/>	<input type="text" value="5"/>	Rispetto alla frequenza media durante le lezioni di questo insegnamento, gli studenti presenti oggi sono...	36

Figura 5.3: La pagina di gestione delle strutture dei questionari

Si può facilmente specificare la posizione e la scala di una domanda per l'anno prescelto inserendole negli appositi campi. Ovviamente si potrà inserire la scala di una domanda solo dopo aver specificato la posizione.

Per non inserire affatto una domanda nella struttura dei questionari di un determinato anno basterà lasciarla con il campo posizione nel valore predefinito “-”.

Questa funzione popolerà solo la tabella STRUTTURAQUESTIONARIO.

5.7 Questionari

Questa funzione popolerà le tabelle SINGOLOQUESTIONARIO e RISPOSTAQUESTIONARIO tramite il caricamento di un file CSV. Il formato di tale file è specificato nella **Tabella 5.5**.

Scl stud.	Scl	ID-CSV	IDINS.	Desc.	Sede	resp1	Anno Isc.	altre risp.		
12	12	25	121640	Moda	Rimini	1	1	8	...	2
12	12	23	120057	Storia	Bologna	2	2	9	...	3
12	12	23	120057	Storia	Bologna	4	1	8	...	4

Tabella 5.5: Il formato del file dei questionari

Durante il caricamento verrà creato, per ogni riga di questo file, un record nella tabella SINGOLOQUESTIONARIO contenente tutte le informazioni che non siano risposte e un record in RISPOSTAQUESTIONARIO per ogni risposta.

5.8 Scarti

Nel caso alcune righe contengano informazioni non valide per l'inserimento, ad esempio valori nulli su campi non opzionali, esse verranno inserite in un file CSV contenente gli scarti del caricamento.

Questo file avrà lo stesso formato dei file caricati tranne per una colonna in più contenente la motivazione dello scarto. Sarà quindi possibile correggere le righe in questione, eliminare la colonna delle motivazioni e ricaricare il file.

Capitolo 6

Conclusioni

Il sistema sviluppato offre un approccio integrato all'interpretazione dei dati sulla qualità della didattica e la soddisfazione degli studenti.

Dopo l'integrazione sui server di ateneo si potrà passare ad un beta test da parte di studenti, docenti e personale amministrativo.

Dato che il sistema è stato costruito su una base di dati flessibile e scalabile potranno essere aggiunte, dopo la fase di beta test, tutte le funzioni che gli stakeholder riterranno opportune.

Parte III
Appendici

Appendice A

Script di creazione del DB

A.1 Tabella ANNO

```
CREATE TABLE ANNO
(
    IDANNO  INTEGER  NOT NULL ,
    ANNOI   INTEGER  NULL ,
    ANNOF   INTEGER  NULL
)
/
CREATE UNIQUE INDEX XPKANNO ON ANNO
(
    IDANNO  ASC
)
/
ALTER TABLE ANNO
    ADD CONSTRAINT XPKANNO PRIMARY KEY (IDANNO)
/
ALTER TABLE ANNO
    ADD CONSTRAINT CK_ANNO UNIQUE (ANNOI , ANNOF)
/
```

Listato A.1: Script di creazione della tabella ANNO

A.2 Tabella DOMANDA

```
CREATE TABLE DOMANDA
(
    IDDOMANDA INTEGER NOT NULL,
    DOMANDA VARCHAR2(2000) NULL,
    DATEINS DATE DEFAULT SYSDATE
)
/
CREATE UNIQUE INDEX XPKDOMANDA ON DOMANDA
(
    IDDOMANDA ASC
)
/
ALTER TABLE DOMANDA
    ADD CONSTRAINT XPKDOMANDA PRIMARY KEY (IDDOMANDA)
/
```

Listato A.2: Script di creazione della tabella DOMANDA

A.3 Tabella INSEGNAMENTO

```
CREATE TABLE INSEGNAMENTO
(
    IDINSEGNAMENTO INTEGER NOT NULL,
    NOMEINSEGNAMENTO VARCHAR2(250) NULL,
    DATEINS DATE DEFAULT SYSDATE
)
/
CREATE UNIQUE INDEX XPKINSEGNAMENTO ON INSEGNAMENTO
(
    IDINSEGNAMENTO ASC
)
/
ALTER TABLE INSEGNAMENTO
    ADD CONSTRAINT XPKINSEGNAMENTO PRIMARY KEY (
        IDINSEGNAMENTO)
/
```

Listato A.3: Script di creazione della tabella INSEGNAMENTO

A.4 Tabella STRUTTURAQUESTIONARIO

```
CREATE TABLE STRUTTURAQUESTIONARIO
(
  IDSTRUTTURAQUESTIONARIO INTEGER NOT NULL,
  IDANNO INTEGER NOT NULL,
  POSIZIONE INTEGER NOT NULL,
  IDDOMANDA INTEGER NOT NULL,
  SCALA INTEGER NOT NULL
)
/
ALTER TABLE STRUTTURAQUESTIONARIO
  ADD CONSTRAINT XPKSTRUTTURAQUESTIONARIO PRIMARY KEY (
    IDSTRUTTURAQUESTIONARIO)
/
ALTER TABLE STRUTTURAQUESTIONARIO
  ADD (CONSTRAINT XFKIDANNO FOREIGN KEY (IDANNO)
    REFERENCES ANNO(IDANNO))
/
ALTER TABLE STRUTTURAQUESTIONARIO
  ADD (CONSTRAINT XFKDDOMANDA FOREIGN KEY (IDDOMANDA)
    REFERENCES DOMANDA(IDDOMANDA))
/
```

Listato A.4: Script di creazione della tabella STRUTTURAQUESTIONARIO

A.5 Tabella RISPOSTAQUESTIONARIO

```
CREATE TABLE RISPOSTAQUESTIONARIO
(
  IDRISPOSTAQUESTIONARIO INTEGER NOT NULL,
  IDSINGOLOQUESTIONARIO INTEGER NOT NULL,
  IDSTRUTTURAQUESTIONARIO INTEGER NOT NULL,
  RISPOSTA INTEGER NOT NULL,
  DATEINS DATE DEFAULT SYSDATE
)
/
ALTER TABLE RISPOSTAQUESTIONARIO
```

```

        ADD CONSTRAINT XPKRISPOSTAQUESTIONARIO PRIMARY KEY (
            IDRISPOSTAQUESTIONARIO)
    /
ALTER TABLE RISPOSTAQUESTIONARIO
    ADD (CONSTRAINT XFKRISPOSTAQUESTIONARIO FOREIGN KEY (
        IDSTRUTTURAQUESTIONARIO) REFERENCES
        STRUTTURAQUESTIONARIO(IDSTRUTTURAQUESTIONARIO))
    /

```

Listato A.5: Script di creazione della tabella RISPOSTAQUESTIONARIO

A.6 Tabella SINGOLOQUESTIONARIO

Nel seguente script di creazione vi è un ALTER TABLE su RISPOSTAQUESTIONARIO poiché non era possibile creare quest'ultimo prima che la tabella SINGOLOQUESTIONARIO esistesse.

```

CREATE TABLE SINGOLOQUESTIONARIO
(
    IDSINGOLOQUESTIONARIO INTEGER NOT NULL,
    IDFACOLTA INTEGER NOT NULL,
    IDCORSODWH INTEGER NOT NULL,
    IDINSEGNAMENTO INTEGER NOT NULL,
    ANNOISCRIZIONE INTEGER NOT NULL,
    DATEINS DATE DEFAULT SYSDATE
)
/
ALTER TABLE SINGOLOQUESTIONARIO
    ADD CONSTRAINT XPKSINGOLOQUESTIONARIO PRIMARY KEY (
        IDSINGOLOQUESTIONARIO)
    /
ALTER TABLE SINGOLOQUESTIONARIO
    ADD (CONSTRAINT XFKSINGOLOQUESTIONARIOINS FOREIGN KEY (
        IDINSEGNAMENTO) REFERENCES INSEGNAMENTO(IDINSEGNAMENTO
        ))
    /
ALTER TABLE RISPOSTAQUESTIONARIO
    ADD (CONSTRAINT XFK2RISPOSTAQUESTIONARIO FOREIGN KEY (
        IDSINGOLOQUESTIONARIO) REFERENCES SINGOLOQUESTIONARIO(
        IDSINGOLOQUESTIONARIO))

```

/

Listato A.6: Script di creazione della tabella SINGOLOQUESTIONARIO

A.7 Tabella SCUOLA_FACOLTA

Nel seguente script di creazione vi è un ALTER TABLE su SINGOLOQUESTIONARIO poiché non era possibile creare quest'ultimo prima che la tabella SCUOLA_FACOLTA esistesse.

```
CREATE TABLE SCUOLA_FACOLTA
(
    IDSCUOLA    INTEGER    NOT NULL ,
    NOMESCUOLA  VARCHAR2(200) NULL
)
/
CREATE UNIQUE INDEX XPKSCUOLA_FACOLTA ON SCUOLA_FACOLTA
(
    IDSCUOLA    ASC
)
/
ALTER TABLE SCUOLA_FACOLTA
    ADD CONSTRAINT XPKSCUOLA_FACOLTA PRIMARY KEY (IDSCUOLA)
/
ALTER TABLE SINGOLOQUESTIONARIO
    ADD (CONSTRAINT XFKSINGOLOQUESTIONARIOFAC FOREIGN KEY (
        IDFACOLTA) REFERENCES SCUOLA_FACOLTA(IDSCUOLA))
/
```

Listato A.7: Script di creazione della tabella SCUOLA_FACOLTA

A.8 Tabella MACROAREA

```
CREATE TABLE MACROAREA
(
    IDMACROAREA INTEGER    NOT NULL ,
    NOMEMACROAREA VARCHAR2(200) NULL
)
```

```
/
CREATE UNIQUE INDEX XPKMACROAREA ON MACROAREA
(
    IDMACROAREA ASC
)
/
ALTER TABLE MACROAREA
    ADD CONSTRAINT XPKMACROAREA PRIMARY KEY (IDMACROAREA)
/
```

Listato A.8: Script di creazione della tabella MACROAREA

A.9 Tabella DIPARTIMENTO

```
CREATE TABLE DIPARTIMENTO
(
    IDDIPARTIMENTO INTEGER NOT NULL,
    NOMEDIPARTIMENTO VARCHAR2(200) NULL
)
/
CREATE UNIQUE INDEX XPKDIPARTIMENTO ON DIPARTIMENTO
(
    IDDIPARTIMENTO ASC
)
/
ALTER TABLE DIPARTIMENTO
    ADD CONSTRAINT XPKDIPARTIMENTO PRIMARY KEY (
        IDDIPARTIMENTO)
/
```

Listato A.9: Script di creazione della tabella DIPARTIMENTO

A.10 Tabella SEDE

```
CREATE TABLE SEDE
(
    IDSEDE INTEGER NOT NULL,
    NOMESEDE VARCHAR2(200) NULL
)
/
```

```
CREATE UNIQUE INDEX XPKSEDE ON SEDE
(
    IDSEDE ASC
)
/
ALTER TABLE SEDE
    ADD CONSTRAINT XPKSEDE PRIMARY KEY (IDSEDE)
/
```

Listato A.10: Script di creazione della tabella SEDE

A.11 Tabella TIPO

```
CREATE TABLE TIPO
(
    IDTIPO INTEGER NOT NULL,
    TIPO VARCHAR2(18) NOT NULL
)
/
CREATE UNIQUE INDEX XPKTIPO ON TIPO
(
    IDTIPO ASC
)
/
ALTER TABLE TIPO
    ADD CONSTRAINT XPKTIPO PRIMARY KEY (IDTIPO)
/
```

Listato A.11: Script di creazione della tabella TIPO

A.12 Tabella CDS__DWH__UNIVOCO

Nel seguente script di creazione vi è un ALTER TABLE su SINGOLO-QUESTIONARIO poiché non era possibile creare quest'ultimo prima che la tabella CDS__DWH__UNIVOCO esistesse.

```
CREATE TABLE CDS_DWH_UNIVOCO
(
    IDDWH INTEGER NOT NULL,
    NOMECDI VARCHAR2(200) NULL,
```

```

        IDTIPO  INTEGER  NOT NULL ,
        IDSEDE  INTEGER  NOT NULL
    )
/
CREATE  UNIQUE  INDEX  XPKCDS_DWH_UNIVOCO  ON  CDS_DWH_UNIVOCO
(
    IDDWH  ASC ,
    IDTIPO  ASC ,
    IDSEDE  ASC
)
/
ALTER  TABLE  CDS_DWH_UNIVOCO
    ADD  CONSTRAINT  XPKCDS  PRIMARY  KEY  (IDDWH)
/
ALTER  TABLE  CDS_DWH_UNIVOCO
    ADD  (CONSTRAINT  XFKTIPO  FOREIGN  KEY  (IDTIPO)  REFERENCES
        TIPO(IDTIPO))
/
ALTER  TABLE  CDS_DWH_UNIVOCO
    ADD  (CONSTRAINT  XFKSEDE  FOREIGN  KEY  (IDSEDE)  REFERENCES
        SEDE(IDSEDE))
/
ALTER  TABLE  SINGOLOQUESTIONARIO
    ADD  (CONSTRAINT  XFKSINGOLOQUESTIONARIODWH  FOREIGN  KEY  (
        IDCORSODWH)  REFERENCES  CDS_DWH_UNIVOCO(IDDWH))
/

```

Listato A.12: Script di creazione della tabella CDS_DWH_UNIVOCO

A.13 Tabella CDS_VARIABILE_ANNUALE

```

CREATE  TABLE  CDS_VARIABILE_ANNUALE
(
    IDDWH  INTEGER  NOT NULL ,
    IDCSV  INTEGER  NOT NULL ,
    IDSCUOLA  INTEGER  NOT NULL ,
    IDMACROAREA  INTEGER  NOT NULL ,
    IDDIPARTIMANTO  INTEGER  NOT NULL ,
    IDANNO  INTEGER  NOT NULL ,

```

```
        DATEINS DATE DEFAULT SYSDATE
    )
/
CREATE UNIQUE INDEX XPKCDS_VARIABILE_ANNUALE ON
    CDS_VARIABILE_ANNUALE
(
    IDDWH ASC,
    IDCSV ASC,
    IDSCUOLA ASC,
    IDMACROAREA ASC,
    IDDIPARTIMANTO ASC,
    IDANNO ASC
)
/
ALTER TABLE CDS_VARIABILE_ANNUALE
    ADD CONSTRAINT XPKCDS_VARIABILE_ANNUALE PRIMARY KEY (
        IDDWH, IDCSV, IDSCUOLA, IDMACROAREA, IDDIPARTIMANTO, IDANNO
    )
/
ALTER TABLE CDS_VARIABILE_ANNUALE
    ADD (CONSTRAINT XFKDWH FOREIGN KEY (IDDWH) REFERENCES
        CDS_DWH_UNIVOCO(IDDWH))
/
ALTER TABLE CDS_VARIABILE_ANNUALE
    ADD (CONSTRAINT XFKSCUOLA FOREIGN KEY (IDSCUOLA)
        REFERENCES SCUOLA_FACOLTA(IDSCUOLA))
/
ALTER TABLE CDS_VARIABILE_ANNUALE
    ADD (CONSTRAINT XFKMACROAREA FOREIGN KEY (IDMACROAREA)
        REFERENCES MACROAREA(IDMACROAREA))
/
ALTER TABLE CDS_VARIABILE_ANNUALE
    ADD (CONSTRAINT XFKDIPARTIMENTO FOREIGN KEY (
        IDDIPARTIMANTO) REFERENCES DIPARTIMENTO(IDDIPARTIMANTO
    ))
/
ALTER TABLE CDS_VARIABILE_ANNUALE
    ADD (CONSTRAINT XFKANNO FOREIGN KEY (IDANNO) REFERENCES
        ANNO(IDANNO))
```

/

Listato A.13: Script di creazione della tabella
CDS_VARIABILE_ANNUALE

A.14 Tabella UTENTE

```
CREATE TABLE UTENTE
(
    IDANAGRAFICAUNIVOCA NUMBER(38) NOT NULL,
    MATRICOLA NUMBER(38) NOT NULL,
    NOME VARCHAR(50),
    COGNOME VARCHAR(50),
    TIPO CHAR(1) DEFAULT 'C',
    DATEINS DATE DEFAULT SYSDATE
)
/
ALTER TABLE UTENTE
    ADD CONSTRAINT XPKUTENTE PRIMARY KEY (
        IDANAGRAFICAUNIVOCA)
/
```

Listato A.14: Script di creazione della tabella UTENTE

A.15 Tabella DOCENTE_INSEGNAMENTO

```
CREATE TABLE DOCENTE_INSEGNAMENTO
(
    IDANAGRAFICAUNIVOCA NUMBER(38) NOT NULL,
    IDINSEGNAMENTO INTEGER NOT NULL,
    IDANNO INTEGER NOT NULL
)
/
CREATE UNIQUE INDEX XPKDOCENTE_INSEGNAMENTO ON
    DOCENTE_INSEGNAMENTO
(
    IDANAGRAFICAUNIVOCA ASC,
    IDINSEGNAMENTO ASC,
    IDANNO ASC
)
/
```

```
/
ALTER TABLE DOCENTE_INSEGNAMENTO
    ADD CONSTRAINT XPKDOCENTE_INSEGNAMENTO PRIMARY KEY (
        IDANAGRAFICAUNIVOCA , IDINSEGNAMENTO , IDANNO)
/
ALTER TABLE DOCENTE_INSEGNAMENTO
    ADD (CONSTRAINT XFKIDANAGRAFICAUNIVOCA FOREIGN KEY (
        IDANAGRAFICAUNIVOCA) REFERENCES UTENTE(
            IDANAGRAFICAUNIVOCA))
/
ALTER TABLE DOCENTE_INSEGNAMENTO
    ADD (CONSTRAINT XFKIDINSEGNAMENTO FOREIGN KEY (
        IDINSEGNAMENTO) REFERENCES INSEGNAMENTO(IDINSEGNAMENTO
        ))
/
ALTER TABLE DOCENTE_INSEGNAMENTO
    ADD (CONSTRAINT XFK2IDANNO FOREIGN KEY (IDANNO)
        REFERENCES ANNO(IDANNO))
/
```

Listato A.15: Script di creazione della tabella
DOCENTE_INSEGNAMENTO

Appendice B

I Tool Utilizzati

Qui di seguito verranno descritti brevemente i tool utilizzati per la realizzazione dell'applicativo.

B.1 Texmaker

Texmaker è uno dei migliori ambienti di sviluppo per documenti \LaTeX .

Vedi **Siti Web Consultati [14]**.

B.2 Gantt Project

Gantt Project è un project management software open source basato su Java compatibile con Windows, Linux e Mac OS X. È dotato di un Diagramma di Gantt per la pianificazione dei task di progetto e di un utility per gestione delle risorse.

Le sue funzioni principali comprendono:

- la creazione di una gerarchia dei task e delle dipendenze;
- la creazione del diagramma Gantt;
- la creazione del grafico del carico delle risorse;
- la creazione del diagramma Pert;

- la creazione di report in PDF e HTML;
- la compatibilità con MS Project;
- lo scambio di dati con applicazioni di gestione delle spreadsheet;
- possibilità di compiere lavoro di gruppo a distanza basato su WebDAV.

Vedi **Siti Web Consultati [10]**.

B.3 WBS Chart Pro

WBS Chart Pro è un project management software basato su Windows. Esso viene utilizzato per creare e mostrare progetti attraverso un diagramma WBS (vedi **Sottosezione 1.3.2**).

Vedi **Siti Web Consultati [15]**.

B.4 Poseidon for UML

Poseidon for UML è un'applicazione software sviluppata dalla Gentleware ed utilizzata per creare modelli con UML (vedi **Sottosezione 2.2.1**). Ha avuto origine dall'ArgoUML project ma erano necessari cambiamenti radicali al fine di rendere ArgoUML un progetto commerciale; come risultato i due sforzi sono molto divergenti.

Vedi **Siti Web Consultati [4]**.

B.5 Eclipse

Eclipse è un progetto open source legato alla creazione di una piattaforma di sviluppo ideata da un consorzio di grandi società quali Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP e Serena Software, chiamato Eclipse Foundation, e creata da una comunità strutturata sullo stile dell'open source.

Vedi **Siti Web Consultati [9]**.

B.6 Hibernate

Hibernate (talvolta abbreviato in H8) è una piattaforma middleware open source per lo sviluppo di applicazioni Java che fornisce un servizio di Object-relational mapping (ORM), ovvero che gestisce la rappresentazione e il mantenimento su database relazionale di un sistema di oggetti Java. Hibernate è stato originariamente sviluppato da un team internazionale di programmatori volontari coordinati da Gavin King; in seguito il progetto è stato proseguito sotto l'egida di JBoss, che ne ha curato la standardizzazione rispetto alle specifiche Java EE.

Vedi **Siti Web Consultati [11]**.

B.7 Shibboleth

Shibboleth è un progetto della Internet2¹ Middleware Initiative. In tale progetto sono state create un'architettura e un'implementazione open-source per la gestione delle identità, nonché per l'autenticazione e l'autorizzazione basate sull'identità federata (o Access Control), funzionante mediante il Security Assertion Markup Language (SAML). L'identità federata permette che le informazioni sugli utenti in un dominio di sicurezza siano fornite ad altre organizzazioni in una federazione. Questo permette un singolo sign on di tipo cross-domain ed elimina la necessità per i fornitori di contenuti di mantenere nomi utente e password. Gli ID Providers (IDP) forniscono le informazioni utente, mentre i fornitori di servizi (SP) utilizzano queste informazioni e consentono l'accesso al contenuto sicuro.

Vedi **Siti Web Consultati [13]**.

¹Internet2 oUCAID (University Corporation for Advanced Internet Development) è un consorzio non-profit che sviluppa tecnologie e applicazioni avanzate per la rete, spesso per trasferimenti ad alta velocità di trasmissione.

B.8 Apache Tomcat

Apache Tomcat (o semplicemente Tomcat) è un contenitore servlet open source sviluppato dalla Apache Software Foundation. Implementa le specifiche JavaServer Pages (JSP) e Servlet di Sun Microsystems, fornendo quindi una piattaforma per l'esecuzione di applicazioni Web sviluppate nel linguaggio Java. La sua distribuzione standard include anche le funzionalità di web server tradizionale che corrispondono al prodotto Apache.

Vedi **Siti Web Consultati [7]**.

B.9 Google Chart API

Le Google Chart API sono delle librerie che permettono di creare grafici all'interno di pagine web.

Vedi **Siti Web Consultati [6]**.

B.10 Oracle

Oracle è uno tra i più famosi database management system (DBMS), cioè sistema di gestione di basi di dati, ed è stato scritto in linguaggio C.

Esso fa parte dei cosiddetti RDBMS (Relational DataBase Management System) ovvero di sistemi di database basati sul Modello relazionale che si è affermato come lo standard dei database dell'ultimo decennio.

Vedi **Siti Web Consultati [12]**.

B.11 PL/SQL Developer

PL/SQL Developer è un software per la gestione del DBMS Oracle sviluppato dalla Allround Automations. Esso permette di creare ed eseguire query, così come di creare e gestire database in ogni loro aspetto.

Vedi **Siti Web Consultati [2]**.

B.12 Toad for Oracle

Analogamente a PL/SQL Developer Toad for Oracle è un software per la gestione del DBMS Oracle. Questo software viene sviluppato dalla ToadSoft.

Vedi **Siti Web Consultati [5]**.

B.13 CA ERwin Data Modeler

CA ERwin Data Modeler (ERwin) è uno strumento di data modeling (analisi dei dati, design di database, etc) di sistemi informativi customizzati che includono basi di dati di sistemi transazionali e data mart. Il motore di data modeling di ERwin è basato sul metodo IDEF1X, anche se ora supporta anche diagrammi mostrati col metodo di information engineering.

Vedi **Siti Web Consultati [8]**.

B.14 FastStone Capture

FastStone Capture è uno strumento di screen capture, potente ma leggero, sviluppato dalla FastStone. Esso permette di catturare facilmente ogni cosa sullo schermo: gli oggetti, i menu, etc. Inoltre permette anche di registrare le attività dello schermo in file video.

Vedi **Siti Web Consultati [3]**.

Appendice C

Tabella di Disambiguazione

Acronimo	Significato
AA	Anno Accademico
CdS	Corso di studio, ad esempio Corso di Studio Scienze di Internet
Dpt	Dipartimento
Ins	Un insegnamento del Corso di Studio nel quale insegnano uno o più professori, ad esempio Programmazione internet
MA	Macroarea
Scl	Scuola
CeSIA	Centro di Sviluppo e Gestione dei Servizi Informatici d'Ateneo
SERED	Statistical Evaluation and Representation of Electronic Data
CSV	Comma-Separated Values
DBMS	DataBase Management System
DB	DataBase
SQL	Structured Query Language
GUI	Graphic User Interface
SSO	Single Sign-On

Tabella C.1: Tabella di disambiguazione

Bibliografia

- [1] Roberto Bruni, Andrea Corradini, and Vincenzo Gervasi. *Programmazione in Java*. Apogeo, 2009.
- [2] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. 2008.
- [3] Gerd Gigerenzer. *Quando i numeri ingannano. Imparare a vivere con l'incertezza*. Cortina Raffaello, 2003.
- [4] Per Kroll and Philippe Kruchten. *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley Professional, 2003.
- [5] Craig Larman. *Applicare UML e i pattern. Analisi e progettazione orientata agli oggetti*. Addison wesley, 2005.
- [6] John Lewis and William Loftus. *Java. Fondamenti di progettazione software*. Addison Wesley Longman Italia, 2001.
- [7] Roger S Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill Science/Engineering/Math, 2010.
- [8] Raghu Ramakrishnan and Johannes Gehrke. *Sistemi di basi di dati*. McGraw-Hill Companies, 2004.

Siti Web Consultati

- [1] Gruppo utilizzatori di $\text{T}_{\text{E}}\text{X}$ e $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. <http://www.guit.sssup.it/>.
- [2] Il sito ufficiale della allround automations. <http://www.allroundautomations.com/>.
- [3] Il sito ufficiale della faststone. <http://www.faststone.org/>.
- [4] Il sito ufficiale della gentleware. <http://www.gentleware.com/>.
- [5] Il sito ufficiale della toadsoft. <http://www.toadsoft.com/>.
- [6] Il sito ufficiale delle google chart api. <http://code.google.com/apis/chart/>.
- [7] Il sito ufficiale di apache tomcat. <http://tomcat.apache.org/>.
- [8] Il sito ufficiale di ca erwin data modeler. <http://erwin.com/products/modeler/>.
- [9] Il sito ufficiale di eclipse. <http://www.eclipse.org/>.
- [10] Il sito ufficiale di gantt project. <http://www.ganttproject.biz/>.
- [11] Il sito ufficiale di hibernate. <https://www.hibernate.org/>.
- [12] Il sito ufficiale di oracle. <http://www.oracle.com/>.
- [13] Il sito ufficiale di shibboleth. <http://shibboleth.net/>.
- [14] Il sito ufficiale di texmaker. <http://www.xmlmath.net/texmaker/>.

- [15] Il sito ufficiale di wbs chart pro. <http://www.criticaltools.com/wbsmain.htm>.
- [16] Wikipedia, l'enciclopedia libera. <http://it.wikipedia.com>.

Elenco delle figure

1.1	Il diagramma delle iterazioni	7
1.2	Il diagramma Gantt del progetto	10
1.3	La lista delle attività stimate con Gantt Project	11
1.4	Il piano di processo nel diagramma Gantt	11
1.5	Il terzo ciclo di iterazione nel diagramma Gantt	12
1.6	La fase di documentazione del progetto nel diagramma Gantt	14
1.7	Il diagramma WBS del progetto	15
2.1	Il diagramma ER relativo al DB che il programma gestirà	19
2.2	Le tabelle relative alla gestione della struttura dell'università	20
2.3	Le tabelle relative alla gestione dei questionari	21
2.4	Le tabelle relative alla gestione degli insegnamenti	22
2.5	Classificazione dei diagrammi UML	27
2.6	Il diagramma dei casi d'uso del terzo ciclo	29
2.7	Il diagramma di attività del terzo ciclo	31
2.8	Il diagramma di stato del terzo ciclo	33
2.9	Il diagramma di sequenza del terzo ciclo relativo all'utente pubblico	35
2.10	Il diagramma di sequenza del terzo ciclo relativo all'utente comune	35
2.11	Prima metà del diagramma di sequenza del terzo ciclo relativo all'utente con privilegi di ROOT	37
2.12	Seconda metà del diagramma di sequenza del terzo ciclo rela- tivo all'utente con privilegi di ROOT	38
2.13	Il diagramma dei componenti del terzo ciclo	39

2.14	Il diagramma delle classi del terzo ciclo	41
2.15	Il package denominato beans	42
2.16	Il diagramma di dispiegamento del terzo ciclo	44
3.1	La home page del sistema vista dall'utente pubblico	49
3.2	Il menù di scelta del CdS	50
3.3	Esempio di istogramma di soddisfazione anonimizzato	51
4.1	Il prospetto docente	53
4.2	Esempio di istogramma di soddisfazione non anonimizzato (per domande)	54
4.3	Esempio di istogramma di soddisfazione non anonimizzato (per insegnamenti)	55
5.1	Il riepilogo	57
5.2	La pagina di gestione delle domande	61
5.3	La pagina di gestione delle strutture dei questionari	62

Elenco delle tabelle

2.1	Tabella delle funzioni	25
5.1	Il formato del file delle afferenze	59
5.2	Il formato del file degli utenti	59
5.3	Il formato del file degli insegnamenti	60
5.4	Il formato del file delle edizioni degli insegnamenti	61
5.5	Il formato del file dei questionari	63
C.1	Tabella di disambiguazione	87

Elenco dei Sorgenti

A.1	Script di creazione della tabella ANNO	69
A.2	Script di creazione della tabella DOMANDA	70
A.3	Script di creazione della tabella INSEGNAMENTO	70
A.4	Script di creazione della tabella STRUTTURAQUESTIONARIO	71
A.5	Script di creazione della tabella RISPOSTAQUESTIONARIO	71
A.6	Script di creazione della tabella SINGOLOQUESTIONARIO .	72
A.7	Script di creazione della tabella SCUOLA_FACOLTA	73
A.8	Script di creazione della tabella MACROAREA	73
A.9	Script di creazione della tabella DIPARTIMENTO	74
A.10	Script di creazione della tabella SEDE	74
A.11	Script di creazione della tabella TIPO	75
A.12	Script di creazione della tabella CDS_DWH_UNIVOCO . . .	75
A.13	Script di creazione della tabella CDS_VARIABILE_ANNUALE	76
A.14	Script di creazione della tabella UTENTE	78
A.15	Script di creazione della tabella DOCENTE_INSEGNAMENTO	78