Università degli Studi di Napoli Federico II

Facoltà di Scienze MM. FF. NN

Corso di Laurea in Informatica



Tesi di Laurea Triennale Sperimentale

Un'interfaccia grafica per la web application VOGCLUSTERS

Relatori

Prof. Guido Russo

Dott. Massimo Brescia

Candidato

Ettore Mancini

Matricola: 566/813

Anno Accademico 2010-2011







Dedicato a mia mamma



SC 🛛 PE

rsità degli Studi di Napoli Federic



Indice

Indice de	elle figure	6	
Indice delle tabelle			
Introduz	ione	9	
1. Il pi	rogetto VOGCLUSTERS	11	
1.1	Descrizione Generale		
1.2	Specifiche Progettuali		
1.3	Architettura generale		
1.4	Stato Iniziale del progetto	16	
2. Il co	ontesto teorico-scientifico		
2.1	Ammassi Globulari		
2.2	Pulsar	19	
2.3	Horizontal Branch Stars	20	
2.4	Text Mining in ambito scientifico	20	
2.5	Virtual Organization	21	
2.6	Il Virtual Observatory	22	
3. Tec	nologie utilizzate	24	
3.1	Metadati e xml		
3.2	Ajax	25	
3.3	GWT		
3.4	Struttura della libreria GWT	29	
3.5	Serializzazione e de-serializzazione di un codice		
3.6	GWT Remote Procedure Call		
3.7	SmartGWT		
4. Des	crizione del sistema		
4.1	Specifiche GUI		
4.2	Pannello login		
4.3	Scheda di benvenuto		
4.4	Struttura grafica generale	40	
4.5	Tab funzionali		
4.6	Sezione di ricerca		
4.6	1 Ricerca per nome	44	





	4.6.2	2	Ricerca generica/parziale per carattere	45
	4.6.3	3	Ricerca per parametro	45
	4.6.4	1	Ricerca per categoria	46
	4.6.5	5	Ricerche avanzate	46
	4.6.6	5	Ricerche Bibliografiche per autore	48
	4.6.7	7	Ricerche Bibliografiche per oggetto	48
4	.7	Bott	oni	48
	4.7.2	1	New Object	49
	4.7.2	2	Update object	52
	4.7.3	3	Delete Object	52
	4.7.4	1	Delete Attribute	53
	4.7.5	5	New Attribute	54
	4.7.6	5	New Biblio	55
	4.7.7	7	New Note	56
	4.7.8	3	New Image	57
	4.7.9	Э	New Plot	58
	4.7.2	10	User List	58
	4.7.2	11	Show my Doc	59
	4.7.2	12	About	59
5.	Impl	lemei	ntazione	61
5	5.1	Casi	d'uso	61
5	5.2	Diag	rammi di classe	63
5	5.3	Stru	ttura codice	64
5	5.4	Punt	o di ingresso del codice	65
5	5.5	Panr	nelli	66
5	5.6	Gest	ione degli eventi	68
5	5.7	Pulsa	anti	68
5	5.8	Forn	٥	69
5	5.9	Fine	stre	69
5	5.10	Tab.		70
5	5.11	Grig	lie	71
5	5.12	Panr	nelli HTML	71
5	5.13	Men	ù a scelta	71





	5.14	Icone	72
	5.15	Remote Procedure Call	72
6.	Risu	ltati	75
	6.1	Login e registrazione	75
	6.2	Home Page	76
7.	Svilu	ıppi futuri	77
8.	Tab	elle di Cockburn	78
9.	Ring	graziamenti	98
10	. В	ibliografia	99







Indice delle figure

Figura 1 - Architettura VOGCLUSTERS	14
Figura 2 - Struttura VOGCLUSTERS	15
Figura 3 - Componenti VOGCLUSTERS	16
Figura 4 - Precedente suddivisione Utenti.	17
Figura 5 - Attuale suddivisione utenti	17
Figura 6 - Cluster	19
Figura 7 - Pulsar	19
Figura 8 - Struttura di una Virtual Organization	21
Figura 9 - Immagine IVOA	22
Figura 10 - Schema delle funzioni dell'IVOA	23
Figura 11 - Collaborazioni IVOA	23
Figura 12 - Differenze Tra Ajax e modello classico	26
Figura 13 - Immagine GWT	27
Figura 14 - Passaggi da Java a Javascript tramite GWT	28
Figura 15 - Plugin di sviluppo da installare sul browser	29
Figura 16 - Layout di default di un progetto GWT	30
Figura 17 - GWT RPC	31
Figura 18 - Struttura client-server SmartGWT	34
Figura 19 - Pannello di avviso GUI	36
Figura 20 - Esempio di tool-tip della GUI	36
Figura 21 - Avviso errore della GUI	36
Figura 22 - Finestra di LogIn	37
Figura 23 - Form di registrazione	37
Figura 24 - Errore nella registrazione	38
Figura 25 - E-mail	38
Figura 26 - Form di LogIn	39
Figura 27 - Pannello iniziale di benvenuto	39
Figura 28 - Finestra Quick Guide	40
Figura 29 - Schema organizzativo della GUI.	40
Figura 30 - Nome utente loggato	40
Figura 31 - Pulsante Logout	41
Figura 32 - Logo VOGCLUSTERS	41
Figura 33 - Link utili	41
Figura 34 - Pulsanti funzionalità	41
Figura 35 - Tipologie di ricerca	42
Figura 36 - Tab	42
Figura 37 - Tab in evidenza	42
Figura 38 - Tab nuovo oggetto aperta	43
Figura 39 - Pannello elenco oggetti	43
Figura 40 - Pannello oggetto.	44
Figura 41 - Ricerca per nome	44





Figura 42 - Ricerca parziale	45
Figura 43 - Parametri disponibili.	45
Figura 44 - Operatori disponibili	45
Figura 45 - Ricerca per categoria	46
Figura 46 - Ricerca avanzata	47
Figura 47 - Pannello Advanced	47
Figura 48 - Pannello Complete	47
Figura 49 - Ricerca della nota bibliografica per autore	48
Figura 50 - Ricerca della nota bibliografica per nome oggetto.	48
Figura 51 - Pulsante New Object.	49
Figura 52 - Tipo di oggetto per New Object	49
Figura 53 - Pannello New Object	50
Figura 54 - Cluster di riferimento.	50
Figura 55 - Form inserimento dati di una Pulsar o di una Star	51
Figura 56 - Oggetto Cluster non presente nel DB	51
Figura 57 - Pulsante Update.	52
Figura 58 - Pannello Update	52
Figura 59 - Pulsante Delete Object	52
Figura 60 - Pannello Delete Object	53
Figura 61 - Pulsante Delete Attribute	53
Figura 62 - Pannello Delete Attribute	54
Figura 63 - Pulsante New Attribute	54
Figura 64 - Pannello New Attribute	55
Figura 65 - Pulsante New Biblio	55
Figura 66 - Pannello New Biblio	56
Figura 67 - Pulsante New Note	56
Figura 68 - Pannello New Note	57
Figura 69 - Pulsante New Image	57
Figura 70 - Pannello New Image	57
Figura 71 - Pulsante New Plot.	58
Figura 72 - Pulsante User List.	58
Figura 73 - Lista utenti	58
Figura 74 - Avviso di insufficienti permessi per la visualizzazione.	59
Figura 75 - Pulsante Show my Doc.	59
Figura 76 - Pulsante About.	59
Figura 77 - Finestra About.	60
Figura 78 - Caso d'uso 1	61
Figura 79 - Caso d'uso 2	61
Figura 80 - Caso d'uso 3	62
Figura 81 - Diagramma delle classi.	63
Figura 82 - Struttura codice	64
Figura 83 - Punto di accesso.	65
Figura 84 - Main XML	65
Figura 85 - Schema layout pannelli.	66
	7





Figura 86 - Schema pannelli orizzontali e verticali	67
Figura 87 - Pannello di login	75
Figura 88 - Home page	76
Figura 89 - Pannello risultati	76

Indice delle tabelle

Tabella 1 - Registrazione	79
Tabella 2 - Login	80
Tabella 3 - Ricerca per nome	81
Tabella 4 - Ricerca per parametro	82
Tabella 5 - Ricerca per categoria	83
Tabella 6 - Ricerca avanzata	84
Tabella 7 - Ricerca completa	85
Tabella 8 - Ricerca per carattere	86
Tabella 9 - Ricerca bibliografica per autore	87
Tabella 10 - Ricerca bibliografica per nome oggetto	88
Tabella 11 - Creazione nuovo oggetto	89
Tabella 12 - Aggiornamenti parametri	90
Tabella 13 - Cancellazione Oggetto	91
Tabella 14 - Creazione nuovo attributo	92
Tabella 15 - Cancellazione nuovo attributo	93
Tabella 16 - Aggiunta nuova nota bibliografica	94
Tabella 17 - Aggiunta nuova nota semplice	95
Tabella 18 - Aggiunta nuova immagine	96
Tabella 19 - Visualizzazione lista utenti	97







Introduzione

Il progetto VOGCLUSTERS riguarda una *web application* per *Data* e *Text Mining* nel settore della ricerca su peculiari oggetti astronomici, gli ammassi globulari (*globular clusters*), in grado di fornire alla comunità scientifica un moderno strumento per la raccolta standardizzata degli archivi di dati relativi ad osservazioni astronomiche con i moderni strumenti osservativi (da terra e dallo spazio), per la navigazione ed esplorazione dei dati, per l'aggiornamento dei risultati scientifici e per la documentazione scientifica nel settore d'interesse. E' importante accennare al fatto che nella comunità astrofisica è in essere da diversi anni un progetto, denominato *Virtual Observatory* (VO), (1), avente l'obiettivo di uniformare la rappresentazione e la descrizione dei dati astrofisici (archivi tabulari e di immagini o grafici), attraverso la definizione di standard progettuali ed implementativi. Nel settore degli ammassi globulari, tuttavia, questo VOGCLUSTERS è dotare la propria base dati di connotati perfettamente compatibili con gli standard VO, oltre a raccogliere vari archivi di dati in un unico database uniformato ai registri VO.

L'applicazione è stata sviluppata dal gruppo DAME nel recente passato. Data la sua complessità il lavoro è stato suddiviso in diverse attività di tirocinio e tesi di laurea. Uno degli aspetti fondamentali riguardava la progettazione ed implementazione di una GUI, in grado di rendere facilmente ed efficacemente fruibile la risorsa alla comunità, sfruttando lo stato dell'arte tecnologico e utilizzando risorse di calcolo distribuito capaci di conservare grandi quantità di dati e garantire una rapida e sicura interazione con gli utenti. Pertanto l'intera applicazione è stata costruita e resa disponibile attraverso la GRID del progetto S.Co.P.E. (Sistema Cooperativo per Esperimenti scientifici ad alte Prestazioni), (2), consistente in una griglia computazionale orientata alla risorse realizzata e gestita dalla collaborazione tra l'Università Federico II e la sezione INFN di Napoli.

Il programma DAME riguarda lo sviluppo di una serie di applicazioni e servizi *web-based*, dunque fondati sul paradigma tecnologico *Web 2.0*, dedicati all'analisi ed esplorazione (*Data Mining*) di archivi di dati in ambito astrofisico. La varietà di siffatti progetti si inserisce dunque nella odierna categoria multidisciplinare dell'X-Informatica, basata sull'utilizzo di tecnologie informatiche in ambiente di calcolo distribuito per la gestione efficiente di enormi quantità di dati relativi a varie discipline di *e-science*, dall'astrofisica, alla biologia, alla geofisica etc.

Il mio contributo originale ha riguardato la progettazione e la realizzazione di una *Graphical User Interface* (d'ora in poi GUI) per la *web application* relativa al progetto denominato VOGCLUSTERS. Il lavoro è stato sviluppato durante il tirocinio in collaborazione con il gruppo del programma DAME (*Data Mining & Exploration*) presso l'INAF (Istituto Nazionale di Astrofisica) con sede locale presso l'Osservatorio Astronomico di Capodimonte. Il codice sorgente creato consta di un centinaio di classi, facenti parte di diversi moduli relativi all'interfacciamento tra l'utente finale e l'architettura interna della *web application*. La GUI per VOGCLUSTERS è stata creata basandosi su una dettagliata analisi dei requisiti, con una minuziosa analisi progettuale e successivamente implementata utilizzando le moderne tecnologie basate sul paradigma asincrono di AJAX, (3), e sui *framework* Google Web Toolkit (GWT), (4), e SmartGWT, (5), in grado di esporre all'utente una risorsa web alla stregua di una normale applicazione desktop.



Il presente documento di tesi è stato strutturato nel seguente modo:

UNI VERSITA²degu STUDI di NA POLI FEDERICO II

Nel CAPITOLO 1 viene descritta l'architettura generale, del sistema già implementato in passato con i relativi problemi e limiti riscontrati. Inoltre vengono definiti gli obbiettivi e *requirements* originari, sia concettuali che implementativi.

Nel CAPITOLO 2 si illustrano gli aspetti teorici relativi all'ambito astrofisico ed alle tecnologie di Data e Text Mining impiegate e realizzate nel progetto.

Nel CAPITOLO 3 si analizzano le tecnologie informatiche utilizzate, evidenziandone i vantaggi e le peculiarità che ne hanno motivato la scelta.

Nel CAPITOLO 4 vengono riassunte le specifiche del sistema.

Nel CAPITOLO 5 si illustrano gli aspetti progettuali, con dettagli tecnici basati su schemi UML, casi d'uso, diagrammi delle classi, struttura delle librerie GWT e SmartGWT e interfaccia con altri sistemi interni della web app.

Nel CAPITOLO 6 si evidenziano i risultati ottenuti.

Nel CAPITOLO 7 vengono analizzati i possibili sviluppi futuri, evidenziando le funzionalità e possibilità di riutilizzo in altri ambiti sociali/scientifici del progetto.

Nel CAPITOLO 8 infine sono illustrate le tabelle di Cockburn.





1.Il progetto VOGCLUSTERS

In questo Capitolo viene descritto il progetto nell'ambito del programma DAME, con una sintesi descrittiva del programma DAME e degli obiettivi scientifici del progetto VOGCLUSTERS, evidenziando i limiti finora riscontrati dalla comunità scientifica sull'argomento, da cui scaturisce l'idea del nostro progetto.

1.1 Descrizione Generale

Il progetto VOGCLUSTERS si inserisce nel programma DAME, (6), come una *web application standalone*, ma con possibilità di interoperabilità ed integrazione con gli altri servizi sviluppati dal gruppo. Come già accennato il programma DAME si prefigge l'obbiettivo di sviluppare un innovativo sistema generale, orientato ai servizi e basato sulle tecnologie *Web 2.0*, di infrastrutture di *Data Mining* in ambiente di calcolo distribuito specializzate nell'esplorazione di *Massive Data Sets* (MDS) con metodi di apprendimento automatico derivati dal paradigma del *machine learning*, (7).

VOGCLUSTERS è nato per facilitare l'analisi e l'esplorazione di una peculiare categoria di dati astronomici con un alto grado di complessità, fornendo ai suoi utilizzatori una facile interfaccia per velocizzare la speculazione scientifica e per consentire un alto grado di usabilità anche per utenti non necessariamente esperti in ambito informatico.

I dati astronomici in questione riguardano archivi o database di ammassi globulari, i quali necessitano di una complessa indagine scientifica spesso basata sul confronto e l'analisi dei vari dati tabellari esistenti e attraverso l'indagine degli articoli scientifici del settore su riviste specializzate. Uno dei grossi problemi è dato dalla presenza di numerosi archivi delocalizzati molto spesso carenti in termini di omogeneità e uniformità nella rappresentazione dei dati.

Un altro problema importante è l'interoperabilità tra i vari archivi distribuiti spesso creati grazie a diverse applicazioni dedicate le quali spesso non garantiscono tale funzione. L'idea quindi nasce proprio per cercare di far confluire tutti i dati in un'unica risorsa *user friendly* per aiutare la comunità scientifica ad uniformarsi ed a focalizzare l'attenzione su un unico strumento omnicomprensivo.

Con lo svilupparsi delle nuove tecnologie osservative (telescopi e strumenti di piano focale di nuova generazione) nel campo astrofisico, i dati raccolti continuano a crescere, fino a ottenere decine di *TeraByte* al giorno, impossibili da analizzare in tempi ragionevoli a causa delle limitate capacità umane degli astronomi. E' quindi necessario ideare applicazioni in grado di fornire strumenti automatici per l'investigazione accurata di tutta questa mole di dati. Ciò anche tenendo in debito conto che in campo astronomico i dati sono costantemente aggiornati, rendendo necessario creare uno strumento capace di aggiornare in tempo reale i dati e permetterne la facile visualizzazione e fruibilità alla comunità scientifica.

Occorre tuttavia tener presente anche che la comunità astrofisica non possiede competenze particolari in ambito informatico, provocando spesso problemi di utilizzo delle varie applicazioni create in precedenza. La creazione di un'interfaccia semplice ed intuitiva facilita dunque molto il lavoro di





indagine scientifica. Si è quindi pensato di creare una *web application* che si avvicinasse molto ad una *Desktop Application*, con il vantaggio di permettere l'accesso in qualsiasi parte del pianeta avendo a disposizione solo un *pc* (o anche un semplice *smartphone o tablet*) con connessione ad internet ed un *browser*, senza avere i comuni problemi di installazione e compatibilità causati dai vari sistemi operativi. Non ultimo resta anche il vantaggio di svincolare l'utente finale dalla necessità di possedere in loco particolari sistemi complessi o risorse di calcolo, rendendo quindi indirettamente fruibili reti di calcolo distribuito delocalizzato, senza conoscerne le peculiari caratteristiche e modalità di utilizzo.

1.2 Specifiche Progettuali

Uno dei requisiti principali dell'applicazione è fornire un'interfaccia grafica *user friendly* comprensibile da utenti anche poco esperti in informatica.

I tre pilastri fondamentali su cui, in generale, si basa una GUI sono:

- **Struttura**: suddivisione delle varie funzioni con i relativi criteri d'interazione;
- Reattività: reazione ad eventi con delle azioni pianificate;
- **Visualizzazione**: dotazione di un aspetto piacevole e in grado di facilitare la comprensione degli oggetti presenti;

La GUI è uno strumento reattivo, cioè esegue un'azione solo dopo un input esterno (come ad esempio l'interazione con bottoni) quindi ogni suo componente deve essere chiaro ed inconfondibile.

Nonostante queste regole, all'utilizzatore vengono comunque richieste alcune conoscenze di base per utilizzare VOGCLUSTERS:

- Saper utilizzare un browser;
- Avere i fondamenti della navigazione sul web e sulla gestione di hyper-testi;
- Avere una conoscenza scientifica di base sugli ammassi globulari e gli oggetti ad essi connessi.

I principali requisiti di funzionamento dell'applicazione VOGCLUSTERS sono:

- **Abilitazione dell'utente**: l'utente VOGCLUSTERS deve avere la possibilità di richiedere l'abilitazione attraverso una specifica procedura di registrazione;
- **Visualizzazione dei dati**: l'applicazione deve essere in grado di interrogare il database, recuperare i dati utili e renderli visibili all'utente attraverso l'interfaccia grafica;
- **Navigazione nei dati**: l'utente deve poter accedere alle informazioni relative ad un oggetto effettuando una ricerca per parole chiave all'interno di VOGCLUSTERS;
- **Ricerca per nome di un oggetto**: l'utente deve poter ricercare un ammasso esistente inserendone il nome completo;
- **Ricerca per parametri**: l'utente deve poter inserire il valore di un parametro ed ottenere gli ammassi relativi a quei valori di parametro;





- Cancellazione di un oggetto: l'utente deve poter cancellare un oggetto;
- Modifica dei valori: l'utente deve poter modificare i valori dei parametri di ciascun oggetto;
- **Ricerca parziale per singolo carattere**: l'utente deve poter inserire un numero o una lettera e deve essere visualizzato l'elenco dei nomi degli oggetti aventi quei caratteri inseriti;
- Estendere le funzionalità a Pulsar e Star: le stesse funzionalità per gli ammassi devono essere attivate anche per Pulsar e Star;
- Visualizzazione di funzionalità differenti in base ai permessi dati all'utente: ogni utente deve poter modificare o cancellare i propri oggetti ma non quelli creati da altri utenti;
- **Visualizzazione delle informazioni**: l'utente deve poter visualizzare in un'unica pagina tutte le informazioni relative a un ammasso globulare;
- Visualizzazione delle viste: l'utente deve poter in un'unica pagina una vista di tutti i valori degli ammassi globulari;

I principali requisiti di funzionamento del database per VOGCLUSTERS sono:

- La base di dati è formata dall'unione delle informazioni provenienti da diversi archivi presenti in rete;
- I dati devono essere rappresentati in formato VO-compliant, (8);
- I dati sono informazioni relative ad ammassi globulari galattici, ammassi extragalattici contenenti oggetti standard e non;
- I parametri originari per gli ammassi galattici sono in larga misura corrispondenti agli attributi della tabella di *Harris* del 2003, (9);
- I parametri per gli ammassi extragalattici sono gli attributi presenti nelle tabelle di M31, (10);
- Gli oggetti pulsar devono avere come attributi (se presenti) anche: posizione, massa, massa della compagna (se in presenza di un sistema binario), periodo;
- Le tabelle che contengono informazioni relative alle stelle del ramo orizzontale devono contenere gli attributi presenti nella tabella di *C. Clement*, (11);
- La base di dati deve memorizzare, in ordine cronologico, le modifiche ai parametri di ogni singolo oggetto con l'indicazione dell'identità del responsabile di tale modifica e con l'obbligo, all'atto dell'inserimento, di riportare almeno una referenza scientifica di riferimento;





- La base di dati memorizza i *tag* associati ai vari oggetti di VOGCLUSTERS e le relazioni fra i *tag* e i dati;
- La base di dati deve garantire la sicurezza di accesso con diritti di scrittura a seconda dei permessi dell'utente;
- Tutti gli utenti DAME e più in generale, gli utenti generici della rete, possono "interrogare" la base di dati;
- L'amministratore DAME può inserire/modificare/eliminare i dati all'interno del database;
- L'utente DAME abilitato può inserire, modificare ed eliminare solo i propri dati;

Un'ulteriore requisito futuro riguarda l'implementazione della *web application* su una architettura di tipo GRID.

L'utilizzo dell'applicazione mediante una griglia computazionale orientata alle risorse (GRID) porta ad innumerevoli vantaggi in termini di performance, efficienza, scalabilità e affidabilità. Con un sistema GRID abbiamo la possibilità di elaborare grandi quantità di dati condividendo le varie risorse tramite organizzazioni virtuali (di cui parleremo in seguito).

1.3 Architettura generale

L'architettura di VOGCLUSTERS è costituita da una struttura a tre livelli:

- User Interface;
- Business Logic Layer;
- Data Access Layer;



Figura 1 - Architettura VOGCLUSTERS.



La **User Interface** svolge il compito di organizzare l'interazione tra utente e sistema e permettere l'interazione con il mondo esterno. Essa contiene tutti gli strumenti per visualizzare, modificare e creare dati.

Il componente **Business Logic Layer** rappresenta l'insieme di regole che regolano il funzionamento dell'applicazione, intercettando le richieste provenienti dallo strato di presentazione e le gestisce opportunamente.

Il componente **Data Access Layer** si occupa della permanenza di tutte le informazioni trattate dall'applicazione e conosce le varie modalità per salvarle e leggerle nell'ambito di una sorgente dati. In particolare, il *Data Access Layer* permette al Business *Logic Layer* di ottenere le informazioni sulle quali effettuare le elaborazioni.



Figura 2 - Struttura VOGCLUSTERS.

Possiamo inoltre identificare le seguenti componenti:

UNI VERSITA²DEGLI STUDI DI NA POLI FEDERICO II

- **VOGCACCESS:** rappresenta il livello di *back-end* dell'applicazione e fa parte del *Data Access Layer*;
- VOGERROR: si occupa della gestione degli errori;
- DATATYPE: contiene tutte le strutture sui tipi di dati usati nell'applicazione;
- **SERVER:** si occupa di raccogliere e gestire le richieste degli utenti e di elaborare i dati ricevuti dal database e rappresenta il *Business Logic Layer*;
- **VOGCFINAL:** è la componente che interagisce con l'utente finale. Esso contiene tutte le classi riguardanti la GUI e rappresenta *il Presentation Layer*.







1.4 Stato Iniziale del progetto

Il progetto VOGCLUSTERS è stato originariamente ideato sulla base degli standard di progettazione di un altro progetto del programma DAME, la *web application* DAMEWARE (*Web Application REsource*), (12), specializzata in *Data Mining* con metodi di *machine learning* su grossi archivi di dati. In origine era inoltre prevista la condivisione del registro utenti da parte delle due applicazioni.

Tuttavia, in corso d'opera è stato scelto di rendere VOGCLUSTERS un'applicazione indipendente, in modo da non gravare ulteriormente su quest'ultima e permettere agli utenti interessati di utilizzare solo gli strumenti offerti da VOGCLUSTERS.

VOGCLUSTERS non aveva una GUI ma visualizzava solo i file xml creati dal lato *server*; inoltre molti requisiti sono stati modificati e/o eliminati in base alle nuove esigenze scientifiche peculiari della *web application* VOGCLUSTERS. A causa della suddetta condivisione utenti, non era previsto un sistema di registrazione tramite *e-mail* di attivazione, ma semplicemente un'abilitazione tramite DAMEWARE. Molte nuove funzionalità, di seguito descritte, richieste durante il mio lavoro non esistevano o erano parzialmente implementate. Va comunque precisato che VOGCLUSTERS, essendo un'applicazione facilmente portabile, ha sempre la possibilità di essere integrata in futuro in altre applicazioni. L'applicazione web prevedeva tre diverse categorie di utenti:

- Gli amministratori;
- Utenti registrati;
- Utenti generici.

Le principali differenze tra queste categorie di utenti riguardavano i diritti di manipolazione delle informazioni.

Gli amministratori avevano maggiori privilegi, mentre gli utenti generici potevano visualizzare solo alcune informazioni senza aver alcuna possibilità di modifica, cosa invece permessa agli utenti registrati.





Nella nuova implementazione è stato eliminato l'utente generico, che aveva la possibilità di navigare nell'applicazione passivamente e visualizzare solo alcune funzioni, mentre sono rimasti gli utenti registrati e gli amministratori mantenendo per entrambi i privilegi di accesso già previsti.



Figura 5 - Attuale suddivisione utenti.





2.Il contesto teorico-scientifico

In questo capitolo sono trattati i vari aspetti teorici e scientifici che riguardano il progetto VOGCLUSTERS.

2.1 Ammassi Globulari

Nella nostra e nelle altre galassie sono presenti ammassi stellari che trovano la loro origine in episodi collettivi di formazione stellare. Gli ammassi globulari, (13), sono un insieme sferoidale di stelle che orbita come un satellite intorno al centro di una galassia e sono sorretti al loro interno da una forte gravità; ciò conferisce loro il tipico aspetto sferico e trattiene, al loro centro, una densità di stelle relativamente molto elevata. Il nome di questa categoria di oggetti deriva dal latino *globus*, che significa "globo" o "sfera"; talvolta ci si riferisce a tali oggetti semplicemente con l'appellativo di "ammasso globulare".

Gli ammassi globulari sono piuttosto numerosi: se ne conoscono attualmente poco più di un centinaio intorno alla Via Lattea, con forse altri 10-20 da scoprire, essendo nascosti all'osservazione da Terra dalle polveri interstellari che oscurano la vista in direzione del centro galattico; si ipotizza che galassie più grandi possano averne un numero nettamente superiore (la Galassia di Andromeda potrebbe averne fino a 500), mentre alcune galassie ellittiche giganti (come M87) ne contano fino a 10.000.

Gli ammassi globulari sono composti generalmente da centinaia di migliaia di stelle vecchie a bassa "metallicità"¹; alcuni sono visibili a occhio nudo e si presentano come delle piccole macchie chiare e dai contorni sfumati. I più luminosi sono Omega Centauri e 47 Tucanae, visibili solo dall'emisfero australe, e l'Ammasso Globulare di Ercole visibile dall'emisfero boreale. Gli studiosi non sono sicuri che le stelle si siano formate in una singola generazione, (14) o se si estendano per diverse generazioni in periodi di varie centinaia di milioni di anni. Questo periodo di formazione stellare è tuttavia relativamente breve se paragonato all'età di molti ammassi. Le osservazioni mostrano che la formazione delle stelle degli ammassi globulari avviene innanzitutto in regioni dove questo fenomeno è molto elevato e dove il mezzo interstellare ha una densità maggiore rispetto alle regioni normali di formazione stellare. La formazione dei suddetti ammassi globulari avviene principalmente nelle regioni dette *starburst*².

Nel 1771 l'astronomo francese *Charles Messier* creò un catalogo, (15), in cui gli ammassi erano indicati dalla lettera M seguita dal numero del catalogo. Una più completa classificazione degli ammassi delle Galassie è quella fornita nel 1888 dal *New General Catalogue*, (16), di galassie, ammassi e nebulose. In essa sono anche riportati numerosi ammassi appartenenti alle due vicine galassie irregolari note come Piccola e Grande Nube di Magellano. Per fare riferimento agli oggetti di questo catalogo si utilizza la sigla NGC seguita dal numero di catalogo. A seguito delle numerose identificazioni, molti oggetti celesti, e in particolare molti ammassi stellari, hanno una molteplicità di nomi variamente e alternativamente utilizzata nella letteratura scientifica.

¹ Per **metallicità** s'intende l'abbondanza di elementi più pesanti dell'idrogeno e dell'elio presente negli oggetti celesti, ovvero la chimica superficiale delle stelle.

² Le **starburst** sono regioni in cui vi è un alto tasso di formazione stellare.





Figura 6 - Cluster.

2.2 Pulsar

Una pulsar è una stella di neutroni, (17). Il suo nome, che significava originariamente *sorgente radio pulsante*, deriva dal fatto che essa contiene venti volte più neutroni che protoni. Nelle prime fasi della sua formazione, in cui essa ruota molto velocemente, la sua radiazione elettromagnetica in coni ristretti è osservata sottoforma di impulsi emessi a intervalli molto regolari. Le pulsar hanno una massa pari a quella del Sole, ma sono compresse in un raggio di una decina di chilometri; da ciò consegue che la loro densità è enorme. Il fascio di onde radio emesso dalla pulsar è causato dall'azione combinata del campo magnetico e della sua rotazione.



Figura 7 - Pulsar.





2.3 Horizontal Branch Stars

Una stella è un corpo celeste che brilla di luce propria, (18). In astronomia e astrofisica il termine designa uno sferoide luminoso di plasma che genera energia nel proprio nucleo, attraverso processi di fusione nucleare; tale energia è irradiata nello spazio sotto forma di onde elettromagnetiche e particelle elementari (neutrini), le quali costituiscono il vento stellare. Nell'applicazione sono analizzate le stelle del ramo orizzontale (19), le quali possiedono la caratteristica di bruciare elio nel loro nucleo centrale. Il termine "orizzontale" è dato dal diagramma HR (20), nel quale le stelle si dispongono su una sequenza approssimativamente orizzontale.

2.4 Text Mining in ambito scientifico

In questo paragrafo ci soffermeremo sul concetto di *Text Mining* evidenziando la sua utilità come strumento efficace di estrazione della conoscenza, inserendosi tra le principali discipline della nuova *e*-science.

Nell'era attuale della *e-science* vengono prodotti quotidianamente centinaia di documenti creati in vari formati e varie lingue. Il *Text Mining* (anche chiamato *Text Data Mining*), (19), è una forma particolare di *Data Mining*, in cui i dati consistono in testi in linguaggio naturale, ossia documenti "destrutturati". Il *Data Mining*, (20), ha per oggetto l'estrazione di conoscenza all'interno di grandi quantità di dati tramite l'applicazione di specifici algoritmi auto-adattivi o statistici, in grado di scoprire nuovi punti di vista e correlazioni tra i dati, altrimenti non relazionabili con le tradizionali tecniche di investigazione .

Oggi il Data Mining si basa su due principi:

- **Estrazione**, cioè estrarre informazione implicita o nascosta in dati già strutturati, con tecniche analitiche, per renderla disponibile e direttamente utilizzabile;
- **Esplorazione ed analisi**, eseguita in modo automatico o semiautomatico, su grandi quantità di dati allo scopo di identificare pattern significativi.

Il *Text Mining* unisce la tecnologia del linguaggio con gli algoritmi del *Data Mining*, inoltre, consente di analizzare il contenuto dei documenti memorizzati nei database e permette di eseguire ricerche semantiche, offrendo un valore aggiunto alla comunicazione scientifica.

Un processo di *Text Mining* è suddiviso in tre fasi:

- Indicizzazione: analizzare il testo, selezionare quello di interesse e poi modificarlo o integrarlo a secondo dei casi;
- **Mining**: in questa fase si applica un algoritmo di *machine learning*;
- Valutazione: vengono valutati ed approvati o rifiutati i risultati ottenuti.





2.5 Virtual Organization

Attualmente esistono molte *Virtual Organization* cioè "un gruppo di individui che collaborano attraverso l'uso di servizi online". Nel caso in esame parliamo di vari gruppi di ricercatori, appartenenti a diverse organizzazioni scientifiche che lavorano a un progetto di ricerca comune dovendo quindi accedere a varie risorse in rete. Ciascun membro della *Virtual Organization* deve collaborare con altri soggetti dislocati in varie zone del pianeta ed essere autorizzati a utilizzare solo determinati servizi online.



Figura 8 - Struttura di una Virtual Organization.

Creando dei servizi online per il *Text Mining*, è chiaro l'enorme guadagno apportato alle comunità virtuali, potendo accedere direttamente al medesimo servizio da qualsiasi parte del pianeta. Solitamente la struttura di una *Virtual Organization* è diffusa sul territorio e le sue attività sono svolte sfruttando un sistema di rete che permette spesso la partnership fra più aziende che lavorano anche in ambiti diversi, ma con lo stesso progetto in comune.

Il *Text Mining* gioca un ruolo fondamentale nelle *Virtual Organization*, poiché sono disponibili nel mondo vari articoli scientifici, sia in formato cartaceo sia in formato elettronico e spesso filtrare i contenuti risulta difficile senza un ausilio tecnologico avanzato. Per questo motivo il *Text Mining* si occupa di ricercare ed organizzare documenti, processarli e filtrarli in base alle esigenze delle comunità .





2.6 Il Virtual Observatory

L'osservatorio virtuale ("Virtual Observatory", VO) consiste in un progetto internazionale che si occupa di raccogliere dati astronomici (immagini, logs di missioni, simulazioni, etc...), organizzarli in modo omogeneo e sviluppare degli strumenti interoperabili, che consentano agli astronomi di accedere in modo standardizzato all'enorme mole di informazioni (ricavate da osservazioni e/o simulazioni) a disposizione della comunità scientifica. Il VO semplifica il lavoro degli astronomi professionisti e rappresenta anche uno strumento di valore per la didattica e la divulgazione. Per tutti i soggetti impegnati a promuovere attivamente l'astronomia presso il pubblico, esso è una grande opportunità, permettendo l'accesso a dati astronomici reali, in modo da poter effettuare esperimenti riguardanti le varie discipline dell'astronomia. Al progetto partecipano attivamente ricercatori di tutti gli osservatori astronomici nel mondo e in Italia in particolare gli astronomi e astrofisici dell'Istituto Nazionale di Astrofisica (INAF). Nell'ambito del VO sono stati definiti degli appositi standard informatici, (21), che permettono a diversi archivi astronomici di poter interagire, e tramite gli applicativi software, di poter correlare tali fonti di informazioni eterogenee, a vantaggio della velocità e dell'efficacia della ricerca.

Con il VO si ha la possibilità di fare scienza combinando tutta l'informazione scientifica del mondo e spostando l'astronomia oltre l'era della classica identificazione di oggetti.

Uno dei problemi annosi è la disponibilità di dati osservativi/simulati in formati differenti con etichette (*labels*) non omogenee tra loro. Al fine di adottare uno standard comune e dei protocolli è stato fondato un gruppo: l'IVOA (International *Virtual Observatory Alliance*).



Figura 9 - Immagine IVOA.

L'IVOA è stata costituita nel giugno 2002 con la missione di facilitare il coordinamento internazionale e la collaborazione necessaria per lo sviluppo e la distribuzione di strumenti, sistemi e strutture organizzative necessarie per consentire l'utilizzo internazionale degli archivi astronomici sottoforma di un sistema integrato ed interoperabile con il VO.







Figura 10 - Schema delle funzioni dell'IVOA.

L'unione delle varie organizzazioni sparse in tutto il mondo permette di unificare la ricerca nei vari ambiti scientifici e velocizzare la conoscenza e la ricerca.



Figura 11 - Collaborazioni IVOA.

L'IVOA comprende 19 programmi VO provenienti da Argentina, Armenia, Australia, Brasile, Canada, Cina, Europa, Francia, Germania, Ungheria, India, Italia, Giappone, Russia, Spagna, Regno Unito, gli Stati Uniti e le organizzazioni intergovernative (ESA ed ESO). L'iscrizione è aperta ad altri programmi nazionali e internazionali secondo le linee guida IVOA di partecipazione, (22).

VOGCLUSTERS si pone in questo contesto come un'applicazione d'avanguardia per rendere fruibili servizi su massive data sets tramite comunicazione remota (vantaggi della *web application* su sistemi *desktop* o *client/server*), diventando uno degli strumenti a disposizione del VO per tutti i progetti riguardanti gli ammassi globulari.





3.Tecnologie utilizzate

In questo terzo capitolo descriveremo le tecnologie utilizzate e alcune delle scelte progettuali che riguardano lo sviluppo dell'applicazione, oggetto di questa tesi.

Negli ultimi anni si è avuta un'enorme diffusione delle *web application*, dovuta anche alla possibilità concessa agli utenti di accedere alle applicazioni mediante comuni *browser*. L'applicazione VOGCLUSTERS è stata sviluppata come una *web application* di tipo *Rich Internet Application* (RIA), (23), Essa rappresenta tipi di applicazioni web che possiedono molte funzionalità interattive e dal punto di vista grafico sono simili alle tradizionali applicazioni per computer, senza necessitare di alcuna installazione su disco fisso.

I motivi principali per cui si sviluppano tali applicazioni sono due:

- Abbattimento dei costi di gestione;
- Facilità di utilizzo senza alcuna installazione o configurazione particolare .

La scelta di sviluppare VOGCLUSTERS come una RIA è data dalla necessità di voler un'applicazione interattiva mediante l'utilizzo di un comune *browser*.

3.1 Metadati e xml

Per la gestione dei dati, VOGCLUSTERS utilizza uno dei più comuni linguaggi standard basati su metadati: XML (*eXtensible Markup Language*), (24), strumento gratuito supportato da una vasta gamma di applicazioni software.

Il termine "metadato" (dal greco meta- "oltre, dopo" e dal latino datum: "informazione") indica l'informazione che descrive un insieme di dati.

Il linguaggio XML è un insieme di regole che permettono di progettare file facili da leggere e da creare su qualsiasi computer. Esso utilizza elementi provenienti dalla sintassi HTML, tra cui tag, attributi e contenuto testuale secondo una serie di regole sintattiche. Di seguito riportiamo un documento XML che rappresenta la struttura di un libro:

<?xml version="1.0" ?> <libro titolo="Corso di XML"> <capitolo titolo="Le regole di XML"> <testo>Un documento XML è un documento di testo... </testo> </capitolo> <capitolo titolo="Schemi XML"> <testo>Un documento XML è valido se ...</testo> </capitolo>





3.2 Ajax

AJAX è l'acronimo di *Asynchronous JavaScript and XML*, (3). Si tratta di una tecnica di sviluppo per la realizzazione di applicazioni di tipo RIA. Essa permette di ottenere una risposta più rapida alle pagine web consentendo al client di richiamare informazioni lato *server* in modo veloce e trasparente.

In una pagina web sviluppata in codice HTML un'operazione eseguita dall'utente viene inviata come richiesta al *web server*. In seguito esso può scegliere di bloccare tale richiesta oppure mostrare una nuova pagina con codice identico alla precedente e conseguente spreco di banda. Ad ogni interazione con la pagina web corrisponde una nuova richiesta al *web server*; di conseguenza, l'interfaccia può risultare più lenta del previsto poiché il tempo di reazione dell'applicazione dipende dal tempo di risposta del *web server*.

Le applicazioni AJAX, invece, possono inviare richieste al *web server* per ottenere solo i dati necessari. La riduzione del flusso di dati trasmesso tra il *browser* e il *web server* determina la diminuzione dei tempi di risposta e rende le applicazioni più veloci.

AJAX prevede lo scambio di dati in *background* che consente l'aggiornamento dinamico della pagina web senza doverla ricaricare interamente. In particolare, essendo un'applicazione di tipo asincrono, i dati extra richiesti al *server* sono caricati senza interferire con il comportamento della pagina esistente.

AJAX utilizza una combinazione di tecnologie:

- HTML e CSS per la formattazione dei documenti, (25);
- DOMper la rappresentazione delle pagine, (26);
- XMLHttpRequest, per lo scambio di dati tra client e server in modo asincrono;
- XML e XSL, per la manipolazione di dati.

Le applicazioni web che usano AJAX richiedono *browser* che supportano le tecnologie necessarie: Mozilla Firefox, Google Chrome, Internet Explorer, Opera, Konqueror e Safari.

In conclusione, AJAX ha due caratteristiche fondamentali:

- Capacità di aggiornare solo una parte delle pagine web senza ricaricarla per intero;
- Capacità di effettuare richieste tramite il protocollo HTTP.



Figura 12 - Differenze Tra Ajax e modello classico.

3.3 GWT

Le attuali applicazioni web diventano sempre più complesse e gli utenti finali sono sempre più esigenti. L'interfaccia utente rappresenta una componente critica delle applicazioni web poiché è necessario mantenere grande interattività tra le varie richieste. Per soddisfare tale necessità, Google ha creato uno strumento, *Google Web Toolkit* (GWT) che permette lo sviluppo di applicazioni web *AJAX-based* in linguaggio Java. In fase di compilazione, il codice viene convertito in Javascript, (27), ottimizzato e compatibile con i principali *browser*, inclusi quelli per dispositivi di tipo mobile (ad es. *Android* o *IPhone*). In particolare, uno sviluppatore in linguaggio Java può sfruttare il proprio *know-how* per utilizzare GWT, sfruttando i vantaggi di Javascript come la velocità (il codice prodotto da GWT è altamente ottimizzato e performante) ed il supporto di tutti i *browser*. Google Web Toolkit è stato sviluppato per sistemi dotati dei seguenti requisiti:

- Sun Java 2 Runtime Environment 1.5;
- Sistema Operativo: Windows Vista/XP/2000/7, Mac OS X 10.4+ (Tiger o Leopard), Linux con GTK+ 2.2.1+;
- Hardware: ~100MB di spazio libero, 512MB RAM.

GWT supporta i seguenti browsers:

- Firefox 1.0, 1.5, 2.0, 3.0,3.5 e ultime versioni;
- Internet Explorer 6, 7,8 and 9;
- Safari 2, 3, and 4;
- Chromium e Google Chrome;
- Opera 9.0.









Figura 13 - Immagine GWT.

Il software di sviluppo (Software Development Kit, SDK), (28), GWT fornisce un insieme di "*Application Programming Interface*" (API) implementate in Java e delle classi dedicate (*Widget*). Tali elementi permettono lo sviluppo di applicazioni AJAX in Java e la compilazione del sorgente in JavaScript altamente ottimizzato.

Il debug di applicazioni AJAX viene eseguito utilizzando un qualsiasi ambiente di sviluppo (*Integrated Development Environment*, IDE).

Una delle caratteristiche fondamentali di GWT è il *Deferred Binding*. Tale tecnica permette la generazione di diverse versioni della stessa applicazione personalizzate per essere eseguite in uno specifico *browser*. Ad esempio, se GWT supporta quattro *browser* e l'applicazione è stata sviluppata in quattro lingue, GWT genererà 16 permutazioni diverse dell'applicazione in fase di compilazione.

Il compilatore GWT esegue ottimizzazioni generali su tutto il codice rinominando metodi, rimuovendo codice inutilizzato, ottimizzando le stringhe, e altro ancora. Esso permette anche la segmentazione in più frammenti JavaScript del *download* impostando dei punti di divisione del codice, con conseguente aumento della velocità di avvio.

La struttura standard di un progetto GWT è la seguente:

- **Cartella src**: contiene codice sorgente java (sia client che server, preferibilmente in pacchetti diversi) e definizioni dei moduli GWT (file XML);
- **Cartella war:** contiene sia risorse statiche sia output compilato; un file in formato Web Archives Repository (WAR) rappresenta l'immagine di sviluppo dell'applicazione; esso include anche il file "web.xml" e i file class/jar necessari al codice *server-side*;
- Cartella test folder: cartella opzionale che contiene codice di test JUnit.

I moduli GWT sono tradotti in moduli Javascript e altri *file* collegati che, per essere eseguiti, devono essere invocati da una pagina *web*. Essi rappresentano le unità di configurazione GWT e contengono le informazioni necessarie ai moduli. Un esempio è rappresentato dalla "entry point class" contenuta in file xml della forma <nome_modulo>.gwt.xml.





Le applicazioni sviluppate con GWT possono essere eseguite in due modalità:

- Web mode: è possibile eseguire l'applicazione come una comune applicazione web; il compilatore GWT traduce il codice java in HTML/javascript interpretabile da qualsiasi browser web;
- Hosted mode: è possibile accedere all'Hosted Mode Console che permette di utilizzare tutti gli strumenti di *debug* disponibili nell'ambiente di sviluppo. La versione più recente di GWT rende possibile l'utilizzo di un qualsiasi *browser* a differenza delle versioni antecedenti che utilizzavano un Hosted Browser (un *browser* integrato).

L'architettura di GWT è composta da quattro componenti principali:

- Il **compilatore GWT Java-to-JavaScript**: componente fondamentale che converte il codice Java in codice Javascript ed è necessario quando si esegue un'applicazione in modalità *Web Mode*;
- Il **GWT Hosted Web Browser:** permette di eseguire le applicazioni in modalità Hosted Mode, ossia di eseguire il codice Java nella *Java Virtual Machine* (JVM) senza convertire il codice in linguaggio JavaScript/HTML;
- JRE emulation library, (29): contiene le implementazioni in linguaggio Javascript delle librerie standard Java maggiormente utilizzate ("*java.lang.*" e "java.util.*"*); il suo compito consiste nel verificare la validità del codice e nell'assicurarsi che non siano state utilizzate classi java non traducibili in javascript. In prima istanza il codice viene eseguito in modalità *Hosted Mode* dopodiché passa in *Web Mode* per la visualizzazione;
- **GWT Web UI class library**: è una libreria contenente un insieme di interfacce e classi che permettono di "disegnare" le pagine *web* (ad es. bottoni, *text boxes*, immagini, ecc.).



Figura 14 - Passaggi da Java a Javascript tramite GWT.



Da notare che, per ciascun *browser*, è disponibile un *plugin* per lo sviluppo e per il *debug* dell'applicazione riconosciuto dall'**Hosted Mode Console**.



Figura 15 - Plugin di sviluppo da installare sul browser.

3.4 Struttura della libreria GWT

La struttura di GWT è molto importante affinché il toolkit possa interagire con il **Convention Over Configuration**, un paradigma di programmazione che prevede una configurazione minima (o addirittura assente) per il programmatore che utilizza un ambiente di sviluppo che lo rispetti. Il **Convention Over Configuration** obbliga il programmatore alla configurazione dei soli aspetti che differiscono dalle implementazioni *standard* o che non rispettano particolari convenzioni di denominazione.

Una struttura standard di GWT è composta da:

- Src/NOME PACKAGE/Project.gwt.xml: file che definisce la configurazione del progetto;
- Src/NOME PACKAGE/CLIENT: cartella destinata a tutte le risorse compilate in JavaScript; gli oggetti al suo interno sono traducibili o serializzabili per l'utilizzo sul *browser*;
- **Src/NOME PACKAGE/PUBLIC**: cartella contenente i file che saranno distribuiti anche al client, ma che non richiedono la compilazione e la traduzione in JavaScript;
- Src/NOME PACKAGE/SERVER: contiene il codice *server-side*, il contenuto non è distribuito al client, non è tradotto in JavaScript e non ha restrizioni sulle regole di utilizzo del Java Runtime Environment.

In particolare, il **NOME PACKAGE** può contenere anche più sottocartelle.





Figura 16 - Layout di default di un progetto GWT.

3.5 Serializzazione e de-serializzazione di un codice

La serializzazione, (30), è il processo di trasformazione di un oggetto nella sua rappresentazione binaria, mentre la de-serializzazione è la procedura inversa, cioè la ricostruzione dell'oggetto a partire da un flusso di *byte*.

L'utilizzo di questi meccanismi ha lo scopo di salvare lo stato e le informazioni presenti all'interno di una struttura dati all'interno di un file da trasmettere su una rete.

Affinchè una classe possa utilizzare il meccanismo di serializzazione e de-serializzazione, è necessario "autorizzarla" tramite l'uso di appositi attributi (*serializable*).

In Java gli oggetti sono tutti serializzabili, fatta eccezione per quelli che si riferiscono alle risorse native del sistema. Un oggetto può essere serializzato tramite il metodo *writeObject*, (30), e de-serializzato tramite il metodo *readObject*, (30). Durante questo processo vengono inclusi dei metadati per rappresentare l'oggetto, i quali sono poi riutilizzati per la de-serializzazione.

3.6 GWT Remote Procedure Call

Una *Remote Procedure Call* (RPC), (31), è una funzione implementata dal *server* e invocata da un client. La comunicazione consiste nell'invio e nella ricezione di oggetti serializzati.

Il *framework* GWT RPC rende facile lo scambio di oggetti java su http per i componenti *client* e *server* che fanno parte di applicazioni *web*. Il codice lato *server* invocato dal client è spesso definito come un servizio la cui implementazione si basa sull'architettura delle *servlet* Java.





GWT gestisce la serializzazione degli oggetti Java in invio e ricezione. In particolare, se il client deve invocare un qualsiasi metodo del *server*, è necessario creare un apposito oggetto, *AsyncCallback*, che gestisce la risposta in modo asincrono. Tale meccanismo è generato automaticamente tramite *Deferred Binding*.

Il seguente schema rappresenta la struttura delle GWT RPC:



Figura 17 - GWT RPC.

Gli elementi coinvolti utilizzando GWT RPC sono:

- Il metodo chiamante: servizio che viene eseguito sul server;
- Il codice del client: codice utilizzato per richiamare il servizio sul server;
- Gli oggetti trasmessi: dati scambiati tra il *client* e server.

Il *server* e il client hanno la possibilità di serializzare e deserializzare i dati; in questo modo gli oggetti di tipo "data objects" possono essere trasmessi come testo ordinario.

Al fine di definire l'interfaccia RPC è necessario sviluppare tre componenti:

- StockPriceService: Interfaccia che estende RemoteService ed elenca tutti i metodi RPC;
- **StockPriceServiceImpl**: Classe che estende *RemoteServiceServlet* e implementa l'interfaccia *StockPriceService*;
- StockPriceServiceAsync: Un'interfaccia asincrona richiamata dal codice client.





L'implementazione del servizio deve estendere la *RemoteServiceServlet* e deve implementare l'interfaccia del servizio associato. Si noti che l'implementazione del servizio non implementa anche la versione asincrona dell'interfaccia. Ogni implementazione del servizio è in definitiva una *servlet* che, invece di estendere una HttpServlet, estende una *RemoteServiceServlet*. Quest'ultima gestisce automaticamente la serializzazione dei dati trasmessi tra il client e il server, richiamando il metodo previsto nell'implementazione del servizio.

3.7 SmartGWT

SmartGWT è un *framework* basato su GWT sviluppato da *Isomorphic Software*, (32), che permette di utilizzare una vasta libreria di *Widget* per l'interfaccia utente e di utilizzarli con il lato server per la gestione dei dati. Esso è molto più avanzato rispetto a GWT poiché offre la possibilità di fruire in modo gratuito delle sue risorse. La versione gratuita offre infatti ottimi strumenti di base con molte funzionalità adatte a varie tipologie di progetto. SmartGWT ha una licenza di tipo *copyleft LGPL*, (33). SmartGWT è stabile, ricco di funzionalità e non presenta problemi di compatibilità con versioni precedenti.

Per una rapida panoramica concettuale di SmartGWT analizziamone i vantaggi e gli svantaggi dovuti al suo utilizzo:

- Vantaggi:

- SmartGWT è una libreria LGPL;
- Ricco numero di componenti supportato da più di 250 esempi con codice sorgente;
- Molto attivo nella comunità e nei forum;
- Supporto per la comunicazione GWT-RPC;
- Rapido sviluppo da parte del team di SmartGWT;
- Nessuna installazione di client HTML5 /AJAX.

- Svantaggi:

- Richiede maggiore documentazione. Il sito con gli esempi e i forum non è pienamente sufficiente;
- I tempi di compilazione sono superiori ai due minuti anche per piccole modifiche con macchine poco performanti.

SmartGWT non è solo un'altra libreria *Widget*, (34), ma un'architettura completa. Essa fornisce tutti i controlli per l'interfaccia utente ed offre tutte le interazioni che gli utenti si aspettano dalle applicazioni *desktop*: finestre, schede, griglie, *form*, strutture ad albero, *tooltips*, *drag* & *drop*, procedure guidate e molto altro ancora. Inoltre, non ha la necessità di essere implementato in tutte le sue parti, ma si può scegliere di implementare solo le componenti che vengono ritenute appropriate.

La maggior parte dei *framework* Ajax si concentra principalmente sulla presentazione e visualizzazione dei dati, per lo più di sola lettura, sia locale che tramite XML / JSON. SmartGWT è stato sviluppato tenendo presente l'integrazione lato server. La maggior parte delle applicazioni





aziendali non necessitano solo della presentazione dei dati, ma devono essere in grado di propagare le modifiche apportate dall'utente inviandole al *server*. La maggior parte dei *framework* basati su AJAX possiede un'ottima gestione di visualizzazione, lasciando però all'utente la scelta della gestione dei dati lato *server*.

L'architettura di SmartGWT si estende dal lato *client* al lato *server*, consentendo alle RIA di comunicare in maniera trasparente tra di loro. Esso fornisce un'enorme quantità di servizi e componenti compatibili con AJAX e offre, inoltre, un *framework* lato *server* per le *web application* basate su java.

Il sistema per la gestione dei dati è affidato alla classe *DataSource*, (35), e rappresenta il fondamento dell'architettura di SmartGWT.

Durante lo sviluppo dell'interfaccia utente l'utilizzo di tale classe permette sia la visualizzazione e l'aggiornamento dei dati. Inoltre, riduce drasticamente la ridondanza tra il codice dell'interfaccia utente e il codice lato *server*.

Un *DataSource* descrive semplicemente il *data model* in maniera concettualmente molto simile alla maniera in cui viene utilizzato in SQL per rappresentare lo schema del database. Nel *DataSource* sono definiti i vari campi, i tipi, i vincoli e se necessario, una chiave primaria, o altri vincoli, come per esempio *enum*, etc.

Si può anche dichiarare una chiave nel *DataSource* che può puntare ad un altro *DataSource*, o essere riferimento a se stesso per descrivere una relazione padre-figlio, come un modello di dati che è definito in uno schema SQL. In sostanza un *DataSource* permette di memorizzare i dati dimensionali, ma con un modello di rappresentazione e relazioni molto ricco.

Dopo aver definito un *DataSource* per descrivere il soggetto / classe di dominio, i dati possono essere letti da varie fonti, sia locali che remote: *array*, XML, JSON, dati dal server o dati di test. Il *DataSource* ha un meccanismo di comunicazione che prevede quattro operazioni fondamentali:

- Fetch: Restituisce i record corrispondenti alle richieste effettuate;
- Add: Crea nuovi record nel DataSource con i valori indicati;
- Update: Aggiorna i record nel DataSource con i valori indicati;
- **Remove**: Cancella i record dal *DataSource*.

Inoltre, un *Datasource* permette la creazione di un'interfaccia grafica che aggiorna i dati utilizzando poche righe di codice. In particolare, l'interazione degli oggetti (pulsanti, griglie, *form*, etc.) con i dati lato server è ben definita e descritta in poche righe di codice.

L'impiego di una delle quattro operazioni sopra elencate porta la creazione di un *DataSource Request* (DSRequest) che la rappresenta. A tale richiesta corrisponde una *DataSource Response* (DSResponse) che utilizza diversi approcci per la connessione al data provider.





Figura 18 - Struttura client-server SmartGWT.

I due principali approcci utilizzati sono:

- Server-side integration (Smart GWT Server Framework): le richieste di DataSource al I. server, effettuate tramite browser, e le corrispondenti risposte sono trasmesse come oggetti Java;
- Client-side integration: le richieste di DataSource sono di tipo HTTP e sono ricevute П. direttamente dal server (in java è possibile utilizzare le API Servlet oppure i file di tipo ".jsp" per gestire le richieste). In questo tipo di approccio le risposte sono inviate in formato XML o JSON.

L'utilizzo del DataSource non è obbligatorio, ma, se non impiegato, il programmatore è costretto ad implementare le comunicazioni con il server in altro modo, ritrovandosi in annosi problemi di compatibilità. Il progetto VOGCLUSTERS è stato interamente creato utilizzando le librerie GWT e SmartGWT; in particolare, quest'ultima è stata scelta poiché offre una vasta gamma di Widget utilizzabili per la creazione di una GUI interattiva e ricca di funzionalità.

SmartGWT ha un ottimo supporto d'interoperabilità che permette con estrema facilità di integrare un suo Widget in un GWT container e viceversa. Tuttavia, essendo SmartGWT molto avanzato, non ha molto senso inserire Widget GWT all'interno dei suoi contenitori. Viceversa risulta più utile utilizzare principalmente Widget SmartGWT, anche per ridurre al minimo i problemi di compatibilità. Ricordiamo infatti che SmartGWT, pur essendo derivato da GWT, può avere dei risultati differenti a seconda dei casi, dovuti allo sviluppo differente dei due prodotti, pur ereditando tutte le caratteristiche di GWT.

In conclusione, SmartGWT è un'architettura completa che offre componenti full-optional per l'interfaccia e per la gestione lato server di applicazioni aziendali; esso fornisce un ambiente pulito e chiaro, Object-Oriented, permettendo quindi una facilità di programmazione delle interfacce utente e massimizzando la compatibilità tra i browser.





4. Descrizione del sistema

In questo capitolo viene data una specifica descrizione della GUI in tutte le sue parti, analizzando singolarmente ciascun componente.

4.1 Specifiche GUI

La necessità di creare una GUI per la gestione di un'applicazione di *Text Mining* è fondamentale per velocizzare e rendere fruibile a tutti le sue potenzialità.

A partire dalle specifiche, è stato necessario individuare un'idea della struttura di base della GUI. Era prioritario rendere immediatamente visibili tutte le funzionalità richieste all'apertura della pagina di benvenuto.

Molto spesso gli utenti meno esperti trovano difficoltoso raggiungere determinate funzionalità di un'applicazione. Spesso sono costretti a dover navigare in sotto-menù alla ricerca della funzionalità desiderata. Per questo motivo si è preferito visualizzare un elenco di pulsanti, ciascuno dei quali, in maniera inequivocabile, attiva una singola funzionalità.

In molte applicazioni, per ogni funzionalità, spesso viene aperta una nuova finestra di lavoro, la quale a volte si sovrappone alle altre. Siccome gli utenti a cui è rivolta la *web application* hanno la necessità di confrontare ed analizzare diversi dati astronomici, era necessario pensare a una struttura che permettesse la sovrapposizione dei pannelli di lavoro, permettendo contemporaneamente un facile accesso, senza dover per esempio ridurre ad icona o tornare alla pagina precedente con un apposito pulsante.

Era quindi necessario che l'utente in qualsiasi caso potesse svolgere le operazioni avendo sempre in primo piano tutte le funzionalità disponibili, potendone usufruire contemporaneamente in qualsiasi momento. Da questa necessità nasce l'idea di utilizzare una struttura basata sulle *tab*, che consente di avere sulla stessa pagina tutte le operazioni svolte e di poter rivedere o chiudere (spostandosi solo con il mouse) le operazioni effettuate in precedenza.

Inoltre, era necessario uniformare l'aspetto grafico rispetto alle altre applicazioni del programma DAME. In particolare è stato necessario riutilizzare lo stesso pannello di registrazione, in modo da renderlo esteticamente identico alle altre applicazioni, ma ovviamente con funzionalità diverse.

Un altro punto importante da non sottovalutare è la creazione di sistemi di controllo per prevenire azioni incontrollate, onde preservare l'integrità e la coerenza dei dati. Per questo importante aspetto sono stati creati appositi avvisi che comunicano all'utente se l'operazione è stata effettuata oppure no. I sistemi di controllo verificano che l'utente abbia i permessi per modificare o cancellare determinati parametri; nel caso in cui all'utente manchino i permessi necessari per effettuare determinate operazioni oppure la funzione richiesta non sia ancora presente nella *web application*, il sistema visualizzerà un opportuno avviso.



HOLE	<u> </u>
	Sorry! Feature available in next release
	OK
	OK

Figura 19 - Pannello di avviso GUI.

Per facilitare l'uso del sistema è stata scelta l'integrazione di appositi *tool-tip*, i quali vengono visualizzati con il passaggio del mouse e guidano l'utente ad una corretta compilazione di determinati *form*.

(
EMPTY	example [023]
EMPTY	[059] [059].xx

Figura 20 - Esempio di tool-tip della GUI.

Nel caso in cui l'utente non inserisca i dati correttamente all'interno dei campi obbligatori, il sistema di controllo prevede l'utilizzo di un simbolo in corrispondenza del campo errato.

VOGCLUSTERS Registration Window
VOGCLUSTERS Registration
Name : 🕕
Family Name : 🕖
Email : 🕕
Confirm Email : 🕕

Figura 21 - Avviso errore della GUI.

Di seguito sono illustrate le principali funzionalità offerte da VOGCLUSTERS ed in che modo sia stata creata la GUI per le varie funzioni.

4.2 Pannello login

Per eseguire l'accesso al sistema (login) è utilizzato un pannello dotato di due form in cui devono essere inseriti username e la password utente. Tale pannello è stato creato basandosi sul layout già utilizzato nell'applicazione DAMEWARE.

Sul pannello sono inoltre presenti due pulsanti:

- LogIn: convalida per l'accesso al sistema nel caso l'utente sia già registrato;
- **Register Now**: attiva la funzionalità di registrazione; se l'utente preme tale pulsante appare una finestra in cui inserire i dati (Figura 23);






Figura 22 - Finestra di LogIn.

Inoltre, sul pannello, sono visualizzate alcune righe di testo che illustrano la procedura di registrazione.

VOGCLUSTERS Registration Windo	w	-×
VOGCLUSTERS Registration	on Form	
Name :		
Family Name :		
Email :	Use a valid address	
Confirm Email :		
Country :		
Affiliation :		
Password :	Allowed range [6,16] characters	
Confirm Password :		
Submit		



Il form di registrazione è composto dai seguenti campi obbligatori:

- Name: nome dell'utente;
- Family Name: cognome dell'utente;
- Email: indirizzo *e-mail*;
- Confirm Email: conferma dell'indirizzo e-mail inserito;
- **Country**: paese di origine;
- Affiliation: organizzazione di appartenenza;
- **Password**: password di lunghezza compresa tra 6 e 16 caratteri e senza caratteri speciali;
- **Confirm Password**: conferma della password.



Nel caso in cui l'utente non inserisca il valore all'interno di un campo, oppure inserisca un valore errato, viene visualizzato un pannello di errore, che segnala la mancanza delle informazioni necessarie per la registrazione.

VOGCLUS	STERS Registration Window	-×
VOGC	CLUSTERS Registration Form	
	Name : 🕕	
Famil	ily Name : 🕕	
	Email : 🕖 Use a valid address	
Confi	Warn	×
	Error on field	
P.	[6,16]	-
P2 Su	Confirm O	

Figura 24 - Errore nella registrazione.

Compilando tutti i campi correttamente, l'utente può premere il pulsante "Submit" per richiedere la registrazione. Dopodiché il sistema invia due *e-mail*: la prima è inviata all'utente e contiene il riepilogo dei dati inseriti, mentre la seconda è inviata all'amministratore del sistema con le informazioni dell'utente che ha richiesto l'accesso. Quest'ultima contiene un *link* ("CLICK HERE TO ACTIVATE USER) che, tramite una RPC, abilita l'utente. Tale politica è stata resa necessaria dal fatto che la *web application* si interfaccia con la GRID del progetto S.Co.P.E., in cui la policy di accesso richiede livelli elevati di sicurezza. La possibilità di attivare manualmente le richieste di registrazione, garantisce la facoltà di verificare preventivamente i dati immessi dall'utente.

The user: user@domain has requested a registration at VOGCLUSTERS. Summary Of User: Name: username Surname: userSurname Country: user Country Affilation: University

CLICK HERE TO ACTIVATE USER DAME - Data Mining & Exploration Program

Figura 25 - E-mail.

L'utente, registrato e abilitato all'accesso al sistema, può procedere all'operazione di *login* inserendo nome utente e *password*.





Figura 26 - Form di LogIn.

Il nome utente è visualizzato in chiaro mentre la password tramite il carattere "•".

4.3 Scheda di benvenuto

La scheda ("*tab*") di benvenuto, denominata "MAIN", contiene le informazioni generali della *web application*. È suddivisa in tre parti:

- **Colonna sinistra**: contiene un *plugin* per l'accesso al gruppo "gclusters" creato all'interno del social network *Facebook*, oltre ad alcune informazioni sul gruppo e gli aggiornamenti più recenti;
- **Colonna centrale**: pannello di benvenuto con il nome dell'utente e un pulsante per accedere alla guida (*QUICK GUIDE*) sulle principali funzionalità della *web application*;
- **Colonna destra:** un feed reader RSS (*Really Simple Syndication*) contenente informazioni sui gruppi "*Group Stellar Clusters*" e DAME. L' RSS, (36), è un formato basato su XML adatto alla diffusione di contenuti in continuo aggiornamento su Internet (ad es. *news*).



Figura 27 - Pannello iniziale di benvenuto.





Figura 28 - Finestra Quick Guide.

La Figura 28 rappresenta la finestra visualizzata in seguito alla pressione del pulsante *QUICK GUIDE*. Al suo interno sono descritte brevemente le principali funzionalità della *web application*.

4.4 Struttura grafica generale

La struttura grafica generale della *web application* è rappresentata nella Figura 29.



Figura 29 - Schema organizzativo della GUI.

Tale struttura presenta le seguenti sezioni:

- Cornice: mostra due funzionalità.
 - i. Titolo dell'applicazione seguito dal nome dell'utente loggato in alto a destra;

VOGCLUSTERS APPLICATION -User: admin

Figura 30 - Nome utente loggato.





ii. Pulsante per eseguire il *logout* in alto a sinistra; l'utilizzo di tale pulsante riporta alla pagina iniziale di *login*.



Figura 31 - Pulsante Logout.

• **Logo**: Contiene il logo dell'applicazione;



Figura 32 - Logo VOGCLUSTERS.

• Link: Contiene un menù a tendina che reindirizza a vari *link*.

Manuals	External Resources	Dame Services	Science Cases	Documents •	Info 🔹
User Manual					
🚺 Web page					

Figura 33 - Link utili.

Ciascuno di essi è suddiviso per categoria, essi sono distribuiti nel modo seguente:

- i. **Manuals**: *link* che riportano a manuali utente;
- ii. External Resouces: link a siti riguardanti l'astronomia;
- iii. Dame Services: link ai vari servizi offerti dal gruppo DAME;
- iv. Science Cases: *link* ai vari casi di studio di tipo astronomico;
- v. **Documents**: *link* a pubblicazioni e documenti su progetti;
- vi. **Info**: *link* riguardanti le informazioni sul gruppo DAME.
- **Buttons**: Contiene la lista delle operazioni rappresentata tramite bottoni, ciascuno dei quali con un nome che identifica la funzionalità alla quale è possibile accedere.

New Object	Update Object	Delete Object	Delete Attribute	New Attribute	New Biblio	New Note	New Image	New Plot	User List	Show my Doc	About
------------	---------------	---------------	------------------	---------------	------------	----------	-----------	----------	-----------	-------------	-------

Figura 34 - Pulsanti funzionalità.

Nei paragrafi successivi si descrive in dettaglio ciascuna funzionalità.





• Search: Contiene i vari pannelli per la ricerca, ciascuno di essi circondato da una cornice su cui viene descritta brevemente la tipologia di ricerca. Essi sono posizionati al di sotto della lista dei bottoni e sono sempre visibili.

F	- Object Name	Object Category	Generic/Partial Chars		Biblio Notes by Author	
l	Value : Type* : Y Search	Category : Yearch	Value :	* Search	Author :	BibliobyAuthor
ŀ	- Object Parameter	,	- Advanced Search		Biblio Notes by Object	
1	Parameter*: V Operator*: V	/alue : Search	Advanced Show all		Object :	BibliobyObject

Figura 35 - Tipologie di ricerca.

In questo pannello sono visualizzate le seguenti tipologie di ricerca:

- i. **Object Name**: ricerca per nome;
- ii. **Object Category**: ricerca per categoria;
- iii. Object Parameter: ricerca per parametri;
- iv. Generic/Partial Chars: ricerca per carattere;
- v. Pulsante Advanced: modalità di ricerca avanzata;
- vi. Pulsante Show all: visualizzazione di tutti gli oggetti presenti nel DB;
- vii. Biblio notes by Autor: ricerca di note bibliografiche inserendo il nome degli autori;
- viii. **Biblio notes by Object**: ricerca di note bibliografiche inserendo l'oggetto di riferimento.

Nei paragrafi successivi si descrivono in dettaglio le varie funzionalità.

• **Tabs**: Sono schede che mostrano i risultati delle varie funzionalità; la *tab* di benvenuto (MAIN) è sempre in evidenza e non è possibile chiuderla. Le altre *tab* che vengono aggiunte hanno la possibilità di essere chiuse cliccando sulla " x " alla destra del loro nome.



Figura 36 - Tab.

L'aggiunta di *tab* avviene in modo progressivo da sinistra verso destra e visualizzata in primo piano.



Figura 37 - Tab in evidenza.



4.5 Tab funzionali

La pressione di un pulsante all'interno delle sezioni "*Buttons*" o "*Search*" determina la visualizzazione di una nuova *tab* contenente i risultati dell'operazione richiesta; in particolare, ciascuna *tab* contiene informazioni specifiche in base al tipo di richiesta effettuata.

MAIN	lew Object 🛛	Update Object 🛛	Delete Object	Delete Attribute	New Attibute 🛛 N	ew biblio 🖾 🛛 New No	te 🛛 New image 🖾	User List 🛛 My Docum	ients 🛛 About			
•	·		4		1.1	New	Object					
Object Type	Cluster	v							* - 4			
NEW CLUSTER							· ·	· · · ·				
All fields with * a	are required										-	
Object Name	e*:		•	Alias Name :	EMPTY		Ra*:			Dec *	:	
L	LII : EMPTY			Bil	EMPTY		Reddening	EMPTY		, Hb_vmag	EMPTY	· · ·
Metallicit	ty : EMPTY			· Trh :	EMPTY		Iso :	EMPTY		Helio_distance	EMPTY	
Galcen_distanc	e : EMPTY			X_distance :	EMPTY		Y_distance	EMPTY		Z_distance	EMPTY	
App_distance_mo	d : EMPTY			Vmag :	EMPTY		Abs_vmag	EMPTY		Ub_color	EMPTY	
Bv_cold	or : EMPTY			Vr_color	EMPTY	· .	Vi_color :	EMPTY		Rrlyr_frequency	EMPTY	
Hb_rati	io : EMPTY			Hb_morph_type	EMPTY	1 1 1 1	Spect_type :	EMPTY		Radial_velocity	EMPTY	
Radial_velocity_er	EMPTY			Lsr_radial_velocity :	EMPTY		Central_concentration	EMPTY		Ellipticity	EMPTY	. *
Core_radiu	US : EMPTY			Halfmass_radius :	EMPTY		Tidal_radius :	EMPTY	. '	Log_core_relaxtime	EMPTY	
Log_hmr_relaxtim	me : EMPTY		Ce	ent_surf_brightness :	EMPTY		Log_lum_density :	EMPTY				
New VOBject						•••						

Figura 38 - Tab nuovo oggetto aperta.

E' possibile visualizzare varie tipologie di *tab* e in alcuni casi l'apertura può avvenire dopo aver eseguito un'operazione all'interno della *tab* precedente.

MAIN	Parameter	search 🖾			
Terzan6	-				
NGC6453		· . ·			
NGC6496					
NGC6541		-			
ESO-SC06					
NGC6584					
GLIMPSE01					
NGC6749	· · · •				· ·
NGC6752					
NGC6760	<u> </u>	-			
NGC6779					
Pal10		5			-
Pal11		•			
NGC6838					
NGC6934			-	· · · · ·	
NGC6981			· · ·		· . *
NGC7006	=				• .
NGC7078					1.1.1
NGC7089	+	J			

Figura 39 - Pannello elenco oggetti.





Ad esempio, selezionando uno degli oggetti presenti nella lista in Figura 39 sarà visualizzata una nuova *tab* contenente le informazioni sull'oggetto scelto.

MAIN Paramet	er search 🛛 GLIMPSE01 🛛		
	Galactic Longitude	31.31	×
GLIMPSEU	Galactic Latitude	-0.10	
	Distance form the Sun [kpc]	4.2	
	Distance from Galactic center [kpc]	4.9	
	Distance component X (toward gal center) [kpc]	3.6	=
	Distance component Y (in direction of gal rotation) [kpc]	2.2	
	Distance comp. Z (toward north gal pole) [kpc]	0.0	
	Right Ascension	184849.7	
	Declination	-012950	
	Apparent visual distance modulus	28.15	
	Integrated V magnitude of the cluster	22.24	
	Absolute visual magnitude (cluster luminosity)	-5.91	
	integrated color indices U-B (uncorrected for reddening)		
	integrated color indices B-V (uncorrected for reddening)		
			1

Figura 40 - Pannello oggetto.

4.6 Sezione di ricerca

Le operazioni di ricerca sono suddivise in:

- Ricerca per nome;
- Ricerca generica/parziale per carattere;
- Ricerca per parametro;
- Ricerca per categoria;
- Ricerche avanzate;
- Ricerche bibliografiche per autore;
- Ricerche bibliografiche per oggetto.

4.6.1 Ricerca per nome

La ricerca per nome permette di trovare un oggetto, presente all'interno del database.

Object Name				
Value : NGC 104	Type* :	Cluster	~	Search
Object Parameter		Cluster		
Parameter * : V Op	erator * :	Pulsar Star		Value :

Figura 41 - Ricerca per nome.





In particolare, il pannello è composto da tre elementi:

- Value: indica il nome completo dell'oggetto;
- **Type**: menù contenente tre possibili scelte: **Cluster**, **Pulsar** oppure **Star**. La selezione del tipo è obbligatoria;
- Search button: invia la richiesta al database per la ricerca dell'oggetto specificato.

4.6.2 Ricerca generica/parziale per carattere

La ricerca generica/parziale di un oggetto è eseguita inserendo, ad esempio, una lettera o parte del nome dell'oggetto.

Generic/Partial Chars		
Value : n	÷	Search

Figura 42 - Ricerca parziale.

In particolare, il pannello è composto da due elementi:

- Value: lettera o parte del nome dell'oggetto ricercato;
- **Search button**: invia la richiesta al database per la ricerca dell'oggetto che contiene il carattere inserito.

4.6.3 Ricerca per parametro

La ricerca per parametro è eseguita inserendo un determinato valore di un parametro.

 Object Pa 	rameter		1	
Parameter * :		۷	Operator Value : Se	arch
MAIN	Gal longitude			
	Gal latitude		and the second se	-
-	х		and the second s	
	Y			
	Z			
Regis	ellipticity		e an account or accedi to see wha	

Figura 43 - Parametri disponibili.

l	 Object Paran 	neter						-0
	Parameter * :	Gal longitude 🗸 Operator * :	< less than	~	Value :	5.9	Sear	ch
Ì	MAIN							
	MAIN		< less than					
			> more than					

Figura 44 - Operatori disponibili.





In particolare, il pannello è composto da quatto elementi:

- Parameter: menù contenente i nomi dei vari parametri;
- **Operator**: menù a scelta degli operatori utilizzabili;
- Value: valore del parametro scelto;
- **Search**: invia la richiesta al database per la ricerca dell'oggetto che soddisfa le condizioni selezionate.

4.6.4 Ricerca per categoria

La ricerca per categoria è eseguita scegliendo una determinata tipologia:



Figura 45 - Ricerca per categoria.

Il pannello è composto da due elementi:

- **Category**: Menù a scelta delle tre categorie:
 - **Photometric**: Informazioni sulla luminosità dell'ammasso (luminosità del ramo orrizzontale, magnitudine in varie bande, etc.);
 - **Positional**: Informazioni sulla localizzazione dell'ammasso (longitudine, latitudine galattica, etc.);
 - **Structural**: Informazioni sulle caratteristiche peculiari dell'ammasso (densità centrale, metallicità, etc.).
- Search: invia la richiesta al database per la ricerca dell'oggetto specificato.

Il sistema non è stato ancora implementato *server-side* ma solo a livello GUI, poiché fa parte degli sviluppi futuri.

4.6.5 Ricerche avanzate

Il pannello di ricerca avanzata è composto da due pulsanti:

- Advanced: Apre una nuova *tab* per la ricerca avanzata Figura 47;
- **Show all**: Apre una nuova *tab* con l'elenco completo degli oggetti presenti nel database Figura 48.





Advanced Search —	*	
Advanced Show all		

Figura 46 - Ricerca avanzata.

MAIN	Advanced 🖾			
Match All ()	Match Any Mate	ch None		
NAME	v equal	s (disregard case) 😺		
Filter				
ame	Gal. Longitude	Gal. Latitude	R_sun	R_gc

Figura 47 - Pannello Advanced.

Il pannello *Advanced* è stato implementato solo a livello GUI, il lato *server-side* fa parte degli sviluppi futuri.

NGC104	-				
NGC288	Ξ	· .			
NGC1261					
NGC1851					
NGC362					
Pal1					
AM1					
Eridanus					
Pal2				· .	
NGC1904		•			
NGC2298					
NGC2419		÷ .			
Pyxis				•	
NGC2808		Ĩ.			

Figura 48 - Pannello Complete.

Il pannello *Complete* è implementato e funzionante in ogni sua parte.





4.6.6 Ricerche Bibliografiche per autore

La ricerca bibliografica per autore permette di visualizzare l'elenco delle note bibliografiche. Il sistema non è stato ancora implementato server-side ma solo a livello GUI, poiché fa parte degli sviluppi futuri.

 Biblio Notes by Author 		1. B. C.
Dibilo Notes by Additor		
Author :	BibliobyAuthor	

Figura 49 - Ricerca della nota bibliografica per autore.

Il pannello è composto da due elementi:

- Author: indica il nome dell'autore;
- **BibliobyAuthor**: invia la richiesta al database per la ricerca specificata.

4.6.7 Ricerche Bibliografiche per oggetto

Il sistema di ricerca bibliografica per oggetto, permette di visualizzare l'elenco delle note bibliografiche connesse all'oggetto associato. Il sistema non è stato ancora implementato *server-side* ma solo a livello GUI, poiché fa parte degli sviluppi futuri.

 Biblio Notes 	by Object
Object :	BibliobyObject

Figura 50 - Ricerca della nota bibliografica per nome oggetto.

Il pannello è composto da due elementi:

- **Object**: indica il nome dell'oggetto;
- BibliobyObject: invia la richiesta al database per la ricerca specificata.

4.7 Bottoni

I pulsanti presenti nella web application sono:

- New Object;
- Update Object;
- Delete Object;
- Delete Attribute;
- New Attribute;
- New Biblio;





- New Note;
- New Image;
- New Plot;
- User List;
- Show my Doc;
- About.

4.7.1 New Object

Attiva la funzionalità di inserimento di un nuovo oggetto.

New Object

Figura 51 - Pulsante New Object.

L'utilizzo di questo pulsante abilita l'apertura di una *tab* (Figura 52) che contiene un menù per la scelta del tipo di oggetto da inserire.

Object Type :		~
	Cluster	
	Horizontal Branch Star	
	Pulsar	



Nella scelta del tipo di oggetto possono verificarsi due casi:

 Nel caso di creazione di un nuovo oggetto è mostrato un pannello (Figura 53), nel quale sono presenti i vari campi da compilare. Il pannello di riferimento è formato da trentanove campi, dei quali solo alcuni sono obbligatori (marcati con un asterisco).







MAIN Nev	/ Object 🖾										
				1.1	New Obje	ct					•
Object Type :	Cluster	Select Object									
Object Name *	:		Alias Name	EMPTY		Ra*	:		Dec *		
LI	EMPTY		Bii :	EMPTY		Reddening	EMPTY		Hb_vmag	EMPTY	*
Metallicity	EMPTY		Trh :	EMPTY		Iso	EMPTY		Helio_distance	EMPTY	ī.
Galcen_distance	EMPTY		X_distance :	EMPTY	· · · · · ·	Y_distance	EMPTY		Z_distance	EMPTY	
App_distance_mod	EMPTY		- Vmag	EMPTY		Abs_vmag	EMPTY		Ub_color	EMPTY	
Bv_color	EMPTY		Vr_color :	EMPTY	·	Vi_color	EMPTY		Rrlyr_frequency	EMPTY	
Hb_ratio	EMPTY		Hb_morph_type :	EMPTY		Spect_type	EMPTY	· · ·	Radial_velocity	EMPTY	
Radial_velocity_erro	EMPTY		Lsr_radial_velocity :	EMPTY	Central	concentration	EMPTY		Ellipticity	EMPTY	
Core radius	EMPTY		 Halfmass radius ; 	EMPTY		Tidal radius	EMPTY		Log core relaxtime	EMPTY	
Log hmr relaxtime	EMPTY		Cent surf brightness	EMPTY	Lo	um density	EMPTY				
New VOBject		3									,

Figura 53 - Pannello New Object.

- Nel caso di creazione di una pulsar o di una star viene mostrato un nuovo pannello (Figura 54), dove è richiesto di definire l'oggetto di riferimento. Il pannello è composto da due elementi:
 - **Cluster**: indica il nome dell'oggetto;
 - Proceed: invia la richiesta al database per la ricerca specificata.

MAIN	New Object 🗵	
Object 1	Type : Pulsar	~
- enter the n	ame of the reference CL	USTER -
Clu	ister :	
Proceed		

Figura 54 -	Cluster	di riferimento.
-------------	---------	-----------------

In caso di presenza dell'oggetto cluster all'interno del DB viene mostrato a video un pannello, nel quale inserire i parametri associati all'oggetto scelto (Pulsar o Horizontal Branch Star). I pannelli di riferimento per gli oggetti sono formati da quattro elementi, ciascuno con i propri parametri specifici.





MAIN New Object 🛛	MAIN New Object 🖾
Object Type : Pulsar	Object Type : Horizontal Branch Star enter the name of the reference CLUSTER Cluster : NGC 104 Proceed
Object Name * : Offset * : Period * : New VOBject	Object Name * : Ra * : Dec * : New VOBject

Figura 55 - Form inserimento dati di una Pulsar o di una Star.

Se l'oggetto non è presente nel database il sistema visualizza un avviso, e viene richiesto all'utente di definire prima l'oggetto, mostrando i vari campi da compilare per la sua creazione.

MAIN	Dect III	W	arn		X		
•		÷	CLUSTER NOT NEW	INTO DB PLEASE CONTINUE T	O CREATE		
Object Type : Pu	ılsar 🗸	1		ОК		* -	
enter the name of th	e reference CLUSTER				-	*	,
Cluster : tes	st						۰.
Proceed							
EW CLUSTER	0 0						
All fields with * are re	equired						
Object Name * :		Alias Name	: EMPTY	Ra * :		* Dec *	:
 Lift: 	EMPTY	· Bi	: EMPTY	Reddening	EMPTY	Hb_vmag	EMPTY
Metallicity :	EMPTY	Trh	EMPTY	ISO	EMPTY	Helio_distance	EMPTY
Galcen_distance :	EMPTY	X_distance	: EMPTY	Y_distance	EMPTY	Z_distance	: EMPTY
pp_distance_mod :	EMPTY	Vmag	: EMPTY	Abs_vmag	EMPTY	- Ub_color	EMPTY







4.7.2 Update object

Attiva la funzionalità di aggiornamento di un parametro di un oggetto.

Update	Object
--------	--------

Figura 57 - Pulsante Update.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 58), nella quale è visualizzato un menù contenente i parametri da inserire.

Il pannello di riferimento è composto da cinque elementi:

- i. **Object Type**: tipo di oggetto;
- ii. **Name**: nome dell'oggetto;
- iii. Value: nuovo valore numerico;
- iv. Attribute Value: parametro da modificare;
- v. **Update value**: richiesta di aggiornamento.

MAIN New Object 🛛 Update Object 🖾 Delete Obje	ject 🗵 Delete Attribute 🖾 New Attribute 🖸		
		Update Value Attribute Object	
All fields with * are required Object Type*: Update value Pulsar	Name *:	Volue *:	Attribute Name * : 🛛 🛛 👻

Figura 58 - Pannello Update.

4.7.3 Delete Object

Attiva il funzionamento di cancellazione di un oggetto dal database.

Delete Object

Figura 59 - Pulsante Delete Object.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 60), nella quale è presente un menù dove inserire il nome e il tipo di oggetto da cancellare.





Il pannello di riferimento è composto da tre elementi:

- i. **Object Name**: nome dell'oggetto;
- ii. **Type**: tipo dell'oggetto;
- iii. **Delete Object**: invia la richiesta di cancellazione dell'oggetto.

MAIN	Delete Object 🗵	
	•	
Object I	lame * :	
	Type* : ~	
Delete	Object	
	Cluster Pulsar	
	Star	

Figura 60 - Pannello Delete Object.

4.7.4 Delete Attribute

Attiva il funzionamento per la cancellazione di un attributo.



Figura 61 - Pulsante Delete Attribute.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 62), dove è inserito il nome dell'attributo e dell'oggetto cui è riferito.

Il pannello di riferimento è composto da tre elementi.

- i. **Object Name**: nome dell'oggetto;
- ii. Attribute Name: nome dell'attributo;
- iii. **Delete attribute**: invia la richiesta di eliminazione del nuovo attributo.





MAIN	Delete	e Attribute 🛙	3	
	•			
		· ' .	•	
Object N	lame * :			
Attribute N	Name * :			
Delete a	ttribute			

Figura 62 - Pannello Delete Attribute.

4.7.5 New Attribute

Attiva il funzionamento di creazione un nuovo attributo da associare ad un oggetto.

New Attribute

Figura 63 - Pulsante New Attribute.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 64), dove è necessario inserire il nome dell'oggetto di riferimento, il nome dell'attributo, una breve descrizione (non obbligatoria), l'**UCD** (*Uniform Content Descriptor*)³, *l'attribute datatype* (cioè il tipo di valore *int*, *double*, etc.) e il valore dell'attributo.

Il pannello di riferimento è composto da sette elementi.

- i. **Object id**: nome dell'oggetto;
- ii. **Optional Name description**: nome opzionale da attribuire all'oggetto;
- iii. Attribute Desk: descrizione dell'attributo;
- iv. Uniform content description: valore dell'UCD;
- v. Attribute Datatype: tipo di attributo;
- vi. Attribute Value: valore numerico dell'attributo;
- vii. New Attribute: richiesta di un nuovo attributo.

³ L'UCD è il campo per fornire in modo standard una descrizione circa il contenuto di un certo attributo di una tabella dati.







MAIN New Attibute 🛛							
			1.1	New Attribute Value		1	
All fields with * are required Object Id * :	2	Optional Name Description		Attribute Desk :	· · · · · · · · · · · · · · · · · · ·	Uniform Content Descritpion :	
Attribute Datatype * :	· ·	Attribute Value * :			1.		
New Attribute							

Figura 64 - Pannello New Attribute.

4.7.6 New Biblio

Attiva la funzionalità di immissione di una nuova nota bibliografica.



Figura 65 - Pulsante New Biblio.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 66), la quale contiene i campi da inserire per il titolo, descrizione, *source/URI*, anno, nome dell'oggetto, numero di autori, id autore.

Il pannello di riferimento è composto da otto elementi:

- i. Title: titolo della nota;
- ii. **Description**: breve descrizione della nota bibliografica;
- iii. Source/URI: fonte o il sito di riferimento;
- iv. Year: cl'anno di pubblicazione;
- v. **Object Name**: l'oggetto di riferimento;
- vi. Numbers authors: il numero di autori;
- vii. **idAuthors**: id dell'autore di riferimento;
- viii. New Biblio: richiesta di aggiunta di una nuova nota bibliografica.





MAIN New	biblio 🗵		
•			÷
Title :			
Description :			
		.:	
Source/URI * :			
Year :	*	100	
Object Name * :			*
Numbers authors :			
IdAuthors :			
New Biblio			

Figura 66 - Pannello New Biblio.

4.7.7 New Note

Attiva il funzionamento d'immissione di una nuova nota semplice.



Figura 67 - Pulsante New Note.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 68), nella quale è presente una finestra dove è possibile inserire il soggetto, la descrizione ed il nome dell'oggetto cui attribuire la nota.

Il pannello di riferimento è composto da quattro elementi:

- i. **Subject**: titolo della nota;
- ii. **Description**: breve descrizione della nota;
- iii. **ID object**: nome dell'oggetto a cui si riferisce;
- iv. New note: invia la richiesta di creazione di una nuova nota semplice.



MAIN New	Note 🛛	
•		÷
Subjet * :		
•		
Description :		
ID object * :		
New note		

Figura 68 - Pannello New Note.

4.7.8 New Image

Attiva la funzionalità di inserimento di una nuova immagine all'interno del DB.



L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 70), nella quale si può scegliere di eseguire l'upload da un sito *web* o dal proprio sistema.

Il pannello di riferimento è composto da tre elementi:

- i. Upload from: le possibili fonti da cui caricare l'immagine;
- ii. **Resource Location**: l'indirizzo, dove è presente l'immagine;
- iii. **Upload**: invia la richiesta di upload.

MAIN	New Note 🛛 New	image 🛛	
Upload from: : UI Ha	RI v IRI Iard Disk	Resource Location : Upload	





La funzione è stata implementata solo a livello GUI poichè l'implementazione *server-side* fa parte degli sviluppi futuri.

4.7.9 New Plot

Attiva la funzionalità di visualizzazione di un plot statistico/analitico. La *tab* relativa non è stata ancora creata poiché fa parte degli sviluppi futuri dell'applicazione.

New Plot

Figura 71 - Pulsante New Plot.

Essa riceverà in input i parametri da plottare e la funzione selezionata dall'utente. Se i parametri saranno corretti, la richiesta verrà inviata al componente PSA (*Plotter* Statistico Analitico). Il grafico prodotto verrà restituito a video in una nuova *tab*.

4.7.10 User List

Attiva la visualizzazione della lista degli utenti.



Figura 72 - Pulsante User List.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 73), nella quale è presente una griglia con l'indirizzo e-mail, il nome e cognome dell'utente.

MAIN User List 🛛										
		÷		User Li	st		3		• • •	
smith@gmail.com	1		Will			Smith				
admin			admin			admin				
rossi@mario.it		1.	Mario		1	Rossi	+			
utente@disabilitato.it			james			brown				

Figura 73 - Lista utenti.

Selezionando uno degli utenti all'interno della lista si aprirà automaticamente il *software* di gestione di posta elettronica utilizzato dall'amministratore.





La *tab* è attiva solo per l'amministratore, gli altri non hanno i permessi per accedervi, e questo viene segnalato tramite un avviso (Figura 74).

Note		X
1	You are logged as guest and you can't see userlist. Please login as admin	
	ОК	

Figura 74 - Avviso di insufficienti permessi per la visualizzazione.

4.7.11 Show my Doc

Attiva la funzionalità di visualizzazione dei dati presenti nella cartella utente della persona connessa. La *tab* relativa non è stata ancora creata poiché fa parte di sviluppi futuri dell'applicazione.

Show my Doc

Figura 75 - Pulsante Show my Doc.

4.7.12 About

Attiva la funzionalità di visualizzazione delle note di creazione della web application.



Figura 76 - Pulsante About.

L'utilizzo di questo pulsante abilita l'apertura di una nuova *tab* (Figura 77), che visualizza alcune informazioni sull'applicazione.







MAIN	About 🖾				
		VogcLUSTERS web a	Ppl/cation		*
VOG	CLUSTERS is	a web application desig	ned for the data a	nd text	
	ig on dottono		giobalai oluotoi ol		* dir.
-					: \
For n	ews and docum	entation please visit <u>http://d</u>	ame.dsf.unina.it		*
					19 - 11 - 11 19 - 11 - 11
Com	inht @ 2044 VO		DAM Nanlas Haly		
Сору		GCLUSTERS - DAME PROG	KAW, Naples, Italy		
All rig code,	hts about the n artwork, pictur	naterial of this application (in es, information) are reserved	ncluding text, compu d to DAME Board	ter	
		Technical Support			
· ·		vogclusters AT gmail.c	om	All States in .	
· ·			 ******** 		
		Skype helpdesk		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	· · ·
		Call met		1994 A	
					· · ·
· ·					· . · . · .
				· · · · ·	
· ·					
20					
	· .			· · ·	
					· · · ·
		· · · ·			
				· .	
				A 14	·
			1 A 1 A 1		
C	NAF-OACN			-	
	1. A				
4		111			

Figura 77 - Finestra About.





5.Implementazione

In questo capitolo sono descritti i casi d'uso, i diagrammi delle classi, la struttura delle librerie GWT e SmartGWT ed i metodi per la creazione delle componenti grafiche del sistema. Inoltre verranno analizzati i casi di test.

5.1 Casi d'uso

Uno dei punti fondamentali nel corretto sviluppo di un sistema consiste nella definizione di casi possibili d'interazione dell'utente con il sistema.

Analizziamo brevemente i principali casi d'uso.

In questo primo use case l'utente può effettuare due operazioni: registrazione e logIn (Figura 78).



Figura 78 - Caso d'uso 1.

Nel secondo use case, l'utente può effettuare varie tipologie di ricerca (Figura 79).



Figura 79 - Caso d'uso 2.





Nel terzo *use case* l'utente deve poter eseguire una delle operazioni previste dal sistema; inoltre è presente l'utente chiamato ADMIN il quale può anche visualizzare la lista utenti (Figura 80).



Figura 80 - Caso d'uso 3.



5.2 Diagrammi di classe

Le classi per accedere alla *home page* dell'applicazione sono strutturate nel seguente modo:





La classe *NewLoginPanel* contiene tutti gli oggetti e i metodi per la creazione della pagina di login, tramite il metodo "*onClick*(event)" può richiamare una delle due classi:

- *RegistrationWindow*: esegue la creazione del pannello di registrazione (Figura 23) nel momento in cui l'utente clicca sul tasto "*Register Now*".
- *MainWindow*: crea la finestra dell'*home page* (Figura 30 e Figura 31) se l'utente clicca sul tasto *LogIn*.

La *MainWindow* richiama la classe *MainPanel*, essa rappresenta il pannello di base (Figura 29) al quale sono collegate le seguenti classi:

- **TopHeader**: *link* e il logo dell'applicazione;
- ButtonPanel: elenco pulsanti delle funzionalità;
- SearchBasic: ricerca per nome, per parametro e per categoria;
- **SearchGeneral**: ricerca generica/parziale per carattere, pulsate per la ricerca avanzata e pulsante per mostrare tutti gli oggetti presenti nel database;
- SearchSpecial: ricerche bibliografiche per autore e per nome oggetto;
- **TabPanel**: visualizzazione delle *tab*.





5.3 Struttura codice

Il componente oggetto di questa tesi prende il nome di VOGCFINAL. Esso ha il compito di interagire con il *server* e database.

VOGCFINAL è stato diviso in tre package: CLIENT, SERVER, SHARED.





Nel package **CLIENT** sono presenti le interfacce sincrone e le dichiarazioni dei metodi RPC.

Inoltre sono stati creati dei sottopackage:

- gui: classi che formano l'interfaccia grafica;
- datatype: classi per la gestione dei tipi degli oggetti;
- data: classi per la gestione dei dati.

Nel package **SERVER** sono presenti le implementazioni dei metodi RPC dichiarati nelle interfacce sincrone, descritte nel package **CLIENT**. Esso contiene le classi *XMLGenerator* e *XMLParser* che, rispettivamente, sono utilizzate per la generazione dei documenti da inviare al *server* e per il parsing dei documenti ricevuti dallo stesso.

Inoltre, vi sono altri due sotto package:

- **Highlevel**: il package contiene una classe che possiede i metodi utilizzati per la traduzione in indirizzi **URL**⁴ delle richieste provenienti dalla GUI;
- LowLevel: il package è utilizzato per effettuare le richieste HTTP verso il SERVER ed esegue i metodi *GET, POST, PUT e DELETE* a seconda della richiesta.

Il package **SHARED** contiene la classe che gestisce la configurazione del sistema (path di riferimento, valori di configurazione, etc.).

⁴ Un *Uniform Resource Locator* (**URL**) è un indirizzo che identifica univocamente una risorsa su Internet.





All'interno della struttura è presente il file Main.qwt.xml (Figura 83), nel quale è dichiarata la classe di partenza chiamata MainEntryPoint (Figura 84).



Figura 83 - Punto di accesso.

```
<?xml version="1.0" encoding="UTF-8"?>
```

<!DOCTYPE module PUBLIC "-//Google Inc.//DTD Google Web Toolkit 1.7.0//EN"

```
<module>
   <inherits name="com.google.gwt.user.User"/>
<inherits name="com.smartgwt.SmartGwt"/>
   <!-- Inherit the default GWT style sheet. You can change -->
   < --- the theme of your GWT application by uncommenting -->
   <!-- any one of the following lines. -->
   <!-- <inherits name='com.google.gwt.user.theme.standard.Standard'/> -->
   <!-- <inherits name="com.google.gwt.user.theme.chrome.Chrome"/> -->
   <!-- <inherits name="com.google.gwt.user.theme.dark.Dark"/> -->
<entry-point class="org.vogc.client.MainEntryPoint"/>
   <!-- Do not define servlets here, use web.xml -->
</module>
```

Figura 84 - Main XML.

Punto di ingresso del codice 5.4

}

Il punto d'ingresso per il programma è il metodo onModuleLoad() definito nella classe MainEntryPoint.

```
public void onModuleLoad() {
       NewLoginPanel LoginPanel= new NewLoginPanel();
      LoginPanel.draw();
```

Il metodo è richiamato nella classe che definisce il punto d'ingresso del codice, e la prima operazione che esegue è la creazione del pannello di login/registrazione.





5.5 Pannelli

La GUI è composta da una serie di pannelli. Nel pannello principale è possibile inserire oggetti (menù, pulsanti, etc.) o sovrapporre altri pannelli più piccoli.

Di seguito vene mostrato uno schema di base (Figura 85):



Figura 85 - Schema layout pannelli.

Per ogni pannello deve essere specificata una direzione: orizzontale o verticale.

I gestori dei pannelli per SmartGWT sono implementati nella classe Layout che fornisce quattro sottoclassi:

- Hlayout: Gestisce la posizione e la larghezza degli oggetti inseriti orizzontalmente;
- VLayout: Gestisce la posizione e la larghezza degli oggetti inseriti verticalmente;
- HStack: Gestisce solo la posizione degli oggetti inseriti orizzontalmente;
- VStack: Gestisce solo la posizione degli oggetti inseriti verticalmente.

Nell'esempio seguente è mostrata la creazione di un pannello orizzontale e di uno verticale:

HLayout PannelloOrizzontale = new HLayout();

Il posizionamento degli elementi creati all'interno di un nuovo pannello orizzontale avviene da sinistra verso destra.

VLayout PannelloVerticale= new VLayout();

Il posizionamento degli elementi creati all'interno di un nuovo pannello verticale avviene dall'alto verso il basso.







Figura 86 - Schema pannelli orizzontali e verticali.

In ogni pannello è possibile definire un'altezza e/o lunghezza con i metodi *setHeight()* e *setWidth()*; la larghezza e l'altezza possono essere espressi come numeri intero di pixel:

```
oggetto.setWidth(140);
oggetto.setHeight(40);
```

oppure come una stringa che esprime una percentuale della dimensione del contenitore (l'intera pagina *web*, o un altro componente di SmartGWT). Essa rappresenta un sistema d'ingrandimento dinamico tramite cui è possibile adattare gli oggetti alle varie risoluzioni.

oggetto.setWidth("100%");

La visualizzazione di un pannello avviene tramite il richiamo il metodo:

pannello.draw();

È possibile mostrare o nascondere un oggetto dopo aver eseguito un comando tramite i metodi show e hide.

show();

hide();





L'aggiunta di oggetti su un pannello avviene tramite i metodi:

- addMember(Canvas component): aggiunge un oggetto di tipo Canvas (oggetto dinamico);
- addMember(Canvas component, int position): aggiunge un oggetto di tipo Canvas nella posizione specificata;
- addMember(Widget widget): aggiunge un oggetto di tipo Widget (bottoni, pulsanti, etc.);
- addMember(Widget widget, int position): aggiunge un oggetto di tipo Widget nella posizione specificata;

5.6 Gestione degli eventi

SmartGWT fornisce la logica per gestire gli eventi generati dalle azioni degli utenti.

Ad ogni evento corrisponde un gestore, fra cui i più utilizzati sono:

- **Click**: per menù e bottoni;
- **RecordClick**: per griglie o tabelle;
- **Changed**: per verificare cambiamenti di stato;
- **TabSelected**: per la selezione di *tab*.

Gli elementi di SmartGWT che supportano determinati eventi, implementano un'interfaccia java del tipo:

Has<NOME EVENTO>Handlers

Dove <NOME EVENTO> rappresenta uno degli eventi descritti in precedenza (ad esempio *HasClickHandlers*). Tali interfacce dispongono di metodi per la registrazione degli eventi.

5.7 Pulsanti

Un pulsante viene definito come un oggetto di tipo "Button":

```
private final Button pulsante;
pulsante = new Button("New Object");
....
pulsante.addClickHandler(new ClickHandler() {
      public void onClick(ClickEvent event) {
           .....
      }
});
```





Il pulsante così definito utilizza il metodo "*addClickHandler*" per la gestione degli eventi generati dall'interazione con lo stesso.

5.8 Form

I form sono caselle di testo in cui l'utente può inserire dei dati per interagire con l'applicazione.

La creazione del *form* avviene nel seguente modo:

private DynamicForm form;

Il tipo *DynamicForm* gestisce una collezione di "FormItems" con i relativi metodi. Durante lo sviluppo dell'applicazione sono stati utilizzati i seguenti:

- **TextItem**: *form* per l'inserimento di caratteri;
- PasswordItem: form per l'inserimento di caratteri visualizzati come
- SelectItem: form per la creazione di un menù a scelta;
- StaticTextItem:oggetto per l'inserimento di testo statico;
- IntegerItem: form per l'inserimento di valori numerici;
- ButtonItem: Bottone associato al form;
- HeaderItem: Intestazione di un form;
- FormItemIcon: Icona associata al form.

I *FormItems* posso essere obbligatori o meno. In particolare, se il campo è obbligatorio, è necessario richiamare il metodo:

setRequired(boolean);

dove "*boolean*" deve essere "*true*"; se l'utente non riempie un campo obbligatorio questo metodo ritorna un errore.

Inoltre, è possibile assegnare ai form dei valori di default:

setDefaultValue("INSERIRE USERNAME");

Dichiarato un *Formitems*, è necessario inserirlo all'interno del *Dynamicform* tramite il metodo *setFields*:

setFields(userName);

5.9 Finestre

Le finestre possono essere personalizzate inserendo vari tipi di oggetti, oppure è possibile richiamare le finestre di *default*.

Per creare una finestra personalizzata bisogna estendere una classe di tipo "Window":





Per inserirvi un oggetto è necessario richiamare il metodo:

addItem(Canvas);

Per visualizzare una finestra di default è necessario utilizzare i metodi della classe SC :

- say(): per visualizzare un pannello informativo;
- warn(): per visualizzare un pannello di errore. •

5.10 Tab

Per creare una tab è necessario utilizzare la classe TabSet:

private TabSet nuovaTab

nuovaTab = new Tab();

Tale classe mette a disposizione vari metodi per la definizione della scheda (posizione, allineamento, etc.).

Il metodo *addTab*() permette di aggiungere una nuova *tab* nel pannello appositamente creato.

Di seguito mostriamo un esempio di creazione di una nuova tab:

}

```
void Start() {
      Tab tStart = new Tab("MAIN");
      WelcomeTABpanel startTab = new WelcomeTABpanel();
      tStart.setPane(startTab);
      toptabset.addTab(tStart);
```

Il metodo Start() richiama la classe "WelcomeTABpanel", formata da vari pannelli orizzontali (Figura 27).

Il metodo setPane inserisce all'interno della tab vuota un nuovo pannello.







5.11 Griglie

Le griglie sono delle tabelle contenenti varie informazioni.

Per creare una nuova griglia si utilizza la classe ListGrid:

ListGrid grid = new ListGrid();

Tramite il metodo *setFields* () si specificano i campi da visualizzare all'interno delle griglie, il quale accetta uno o più record di oggetti che definiscono i campi di tale componente.

Per definire i campi all'interno delle griglie è necessario utilizzare il metodo:

grid.setFields(ListGridFields);

dove ListGridFields rappresenta la classe che contiene i nomi dei campi da inserire nella griglia.

Per caricare i dati all'interno delle griglie è necessario utilizzare il metodo:

```
grid.setData(ListGridRecord);
```

dove ListGridRecord rappresenta la classe che contiene i record da caricare nella griglia.

5.12 Pannelli HTML

É possibile creare dei pannelli che accettano porzioni di codice HTML.

HTMLPane htmlPane = new HTMLPane(); String contents = "<iframe src='http://www.facebook.com/plugins/likebox.php?href </iframe> ";

htmlPane.setContents(contents);

Il metodo "setContents" associa il codice in HTML alla classe HTMLPanel .

5.13 Menù a scelta

Il menù a scelta è un tipo di oggetto che visualizza le possibili scelte che l'utente può effettuare.

La dichiarazione dell'oggetto è la seguente:

SelectItem menu = new SelectItem();

Il seguente metodo definisce il contenuto del menù:

```
menu.setValueMap("Cluster", "Pulsar", "Star");
```





Ad ogni scelta presente nel menù, corrisponde un evento che viene gestito tramite il seguente metodo:

```
public void onChanged(ChangedEvent event) {
    ......
}
```

All'interno di tale metodo sono definite delle operazioni che permettono di gestire la scelta da parte dell'utente. Ad esempio, il metodo "*getValue()*" può essere utilizzato per catturare la selezione effettuata.

event.getValue();

5.14 Icone

Le icone delle immagini interattive che permettono di attivare determinate funzioni. Per la creazione di un oggetto di tipo Icona è necessario utilizzare la classe FormitemIcon .

FormItemIcon icon = new FormItemIcon();

Per specificare la path di riferimento dell'immagine si utilizza il seguente metodo:

icon.setSrc("C:\image.png");

Per rendere l'icona interattiva e gestire l'evento corrispondente, è necessario utilizzare il seguente metodo:

```
public void onIconClick(IconClickEvent event) {
    ....
}
```

5.15 Remote Procedure Call

Il meccanismo delle RPC consente l'esecuzione di funzioni su computer remoti. Per l'implementazione vengono definiti tre elementi:

• interfaccia "Service": definisce, in modo sincrono, i metodi da utilizzare per la gestione dei servizi lato client;

```
public interface SearchService extends RemoteService {
    public SearchResult search(String s);
}
```




• interfaccia "Service Asynchronous": definisce, in modo asincrono, i metodi da utilizzare per la gestione dei servizi lato client; in particolare, i metodi sono di tipo "void" e la lista dei parametri di ciascun metodo deve includere un oggetto di tipo "AsyncCallback":

```
public interface SearchServiceAsync {
    public void search(String s, AsyncCallback<SearchResult>
callback);
}
```

• classe "Service Implementation": Implementa i servizi dichiarati nell'interfaccia "Service".

Di seguito mostriamo un esempio di implementazione del servizio di Ricerca:

```
public class SearchServiceImpl extends RemoteServiceServlet implements
SearchService {
      private static WebAppUseCase webapp_uc = new WebAppUseCase();
      private XMLParser xmlp = new XMLParser();
      public SearchResult search(String toSearch) {
             SearchResult src = new SearchResult();
             try {
             ErrorReport report = new ErrorReport();
             String result = webapp_uc.WAsearch(toSearch);
             if (result.equals(Messages.CONNECTION_NOT_FOUND)) {
                   report.setCode(Messages.CONNECTION_NOT_FOUND);
                   report.setMessage(Messages.getErrorString(Messages.CONNE
                   CTION_NOT_FOUND));
             report.setAdditionalInfo("");
             src.setReport(report);
             } else {
                   src = xmlp.parseSearchResult(result);
             }
             } catch (TemplateException ex) {
                   Logger.getLogger(SearchServiceImpl.class.getName()).log(
                   Level.SEVERE, null, ex);
             }
             return src;
      } }
```





Ad ogni funzionalità dell'interfaccia grafica corrisponde la gestione di una RPC; in particolare, le risposte di tali chiamate vengono gestite tramite l'implementazione dell'interfaccia "*AsyncCallback*". Nel caso in cui la chiamata asincrona risponda con successo, viene richiamato il seguente metodo:

Altrimenti, se l'operazione non va a buon fine, viene richiamato il metodo:

public void onFailure(Throwable caught) {

}







6.Risultati

In questo capitolo mostriamo i risultati ottenuti dallo sviluppo dell'applicazione VOGCLUSTERS.

6.1 Login e registrazione

La Figura 87 - Pannello di login. Figura 87 mostra la pagina di accesso all'applicazione. Essa contiene una breve descrizione degli scopi della stessa, i campi in cui inserire le credenziali di accesso e, inoltre, una breve descrizione della procedura di registrazione per i nuovi utenti.



Figura 87 - Pannello di login.





6.2 Home Page

La Figura 88 mostra la "*home page*" dell'applicazione visualizzata dopo aver effettuato l'accesso. In essa sono presenti tutte le funzionalità descritte nei capitoli precedenti.



Figura 88 - Home page.

Le funzionalità sono sempre a disposizione dell'utente e ad ogni richiesta è associata la creazione di una nuova *tab*, visualizzata nella sezione apposita (Figura 89).

VOGCLUSTERS API	PLICATION -User: admin						Logout : 💄
The second se	Manuals	External Resources	 Dame Services 	 Science Cases 	 Documents 	• Info	-
Pes Par							
A NUMBER							
- any and	COCULATION and application						
New Object	Undate Object Delete Object Delete Attribute New Attribute New Bib	In New Note New Imag	New Plot User List St	now my Doc About			
new object	obrase oblect pelese oblect pelese womente Hell womente Hell mo	io nen note nen biley		an in cos			
Object Name	Add Nameda (Physics, 191) (Second)	Object Category	Generic/Partial Chars	(a)	Biblio Notes by Author	(manufacture and and	
value : ngc1	ost type: Cooler M Search	Callegory : Search	Value :	Second	Biblis Notes by Object	BiblioDyAuthor	
Object Parameter			Advanced Show all		Object :	BibliobyObject	
Parameter * : _	Value :	Search				(and) adjust	
MAIN O	ge1851 🖸						
	Right Ascension		78.53	N3			
ngc1851	Declination		-40.0	472			
	Galactic Longitude		244.5	51			
	Galactic Latitude		-35.0	4			
	Galactic Lonpitude		244.5	51			
	Galactic Latitude		-35.0	0			
	Distance form the Sun [kpc]		12.1				
	Distance from Galactic center [kpc]		16.6				
	Distance component X (loward gal center) [kpc]		-4.2				
	Distance component Y (in direction of gal rotation) [kpc]		-8.9				
	Distance comp. Z (toward north gal pole) [kpc]		-6.9				
	Right Ascension		0514	06.76			1
	Declination		-4003	247.6			
	Apparent visual distance modulus		15.47				
	Integrated V magnitude of the cluster		7.14				
	Absolute visual magnitude (cluster luminosity)		-8.33				
	integrated color indices U-B (uncorrected for reddening)		0.17				
	integrated color indices B-V (uncorrected for reddening)		0.76				
	integrated color indices V-R (uncorrected for reddening)		0.49				
	integrated color indices V-I (uncorrected for reddening)		1.01				
	Metallicity: Fe/H [Sun]		-1.18				
	Foreground Reddening		0.02				
	V Magnitude Level of the Horizontal Branch (or RR Lyraes)		16.05	•			
	integrated spectral type		F7				
	elipticity (projected axial ratio)		0.05				
	heliocentric radial velocity [km/s]		320.5	5			
	observational (internal) uncertainty in radial velocity		0.6				

Figura 89 - Pannello risultati.





7. Sviluppi futuri

Come descritto nei paragrafi precedenti, l'applicazione VOGCLUSTERS dovrà essere ottimizzata mediante il completamento e miglioramento di alcune funzionalità e lo sviluppo di ulteriori tecniche di analisi e visualizzazione dei dati.

In particolare, dovranno essere completate le operazioni *server-side* riguardanti la ricerca per categoria, avanzata, bibliografica per autore e per nome dell'oggetto. Inoltre sarà implementato il sistema lato *server* per l'upload delle immagini e ampliata la visualizzazione delle informazioni relative a ciascun ammasso con l'aggiunta di *link*, note bibliografiche ed immagini.

Un ulteriore valore aggiunto all'applicazione sarà lo sviluppo della funzionalità per la creazione di un Plot statistico/analitico. Una prima versione darà la possibilità di eseguire l'*upload* di un file (rigorosamente con estensione **CSV**⁵), e attraverso opportune opzioni creare il *plot* e visualizzarlo in una *tab* del *browser*, senza però prevedere il salvataggio. Un successivo sviluppo permetterà all''utente di selezionare una lista di oggetti dal database, scegliere determinati parametri, eventualmente trasformare gli stessi in file di varie estensioni e fare un plot da poter salvare su *hard disk* o inviarlo a programmi specifici. L'implementazione avverrà attraverso **STILTS**⁶ (39), per cui sarà necessario verificare ed inserire nella *tab* di *setup* tutte le opzioni richieste dal comando della libreria. Sarà infine ampliato il database locale degli oggetti con l'introduzione di tabelle *VO-compliant*, ricavate dal registro del VO.

⁵ **Comma-Separated Values** è un formato di file basato su file di testo utilizzato per l'importazione ed esportazione (ad esempio da fogli elettronici o database) di una tabella di dati.

⁶ Starlink Tables Infrastructure Library Tool Set è un programma a linea di comando per processare dati tabulari.





8. Tabelle di Cockburn

In questo capitolo sono analizzate le tabelle di *Cockburn* (40) che si riferiscono ai possibili casi d'uso (Figura 78, Figura 79, Figura 80), riguardanti la *web application*.

Ciascuna tabella descrive in dettaglio il meccanismo di azione-reazione per ogni interfaccia utentesistema software, in relazione ai vari casi d'uso previsti dall'applicazione. Quindi dettagli circa gli eventi che si presentano all'utente in caso di successo di una determinata operazione, le situazioni di errore, il modo in cui si comporta il sistema e chi è l'attore che può utilizzare tali funzionalità.

Le tabelle di *Cockburn* hanno i seguenti valori:

- Use Case: nome dell'evento;
- Goal in contex: scopo dell'evento;
- Precondition: precondizioni per accedere all'evento;
- Success end condition: condizione per stabilire che l'evento è andato a buon fine;
- Failed end condition: condizione per stabilire che l'evento non è andato a buon fine;
- Primary actor: soggetto principale che esegue le operazioni;
- Trigger: operazione che fa partire l'evento;
- **Step**: singola operazione per l'evento;
- Extension: operazioni alternative ad uno step (opzionali).

SINTESI TABELLE DI COCKBURN							
USE CASE	OGGETTO	RIFERIMENTI TABELLA					
1	Registrazione nuovo utente al sistema	1 – pagina79					
2	Login utente registrato	2 – pagina 80					
3	Ricerca oggetto per nome	3 – pagina 81					
4	Ricerca oggetto per parametro	4 – pagina 82					
5	Ricerca oggetto per categoria	5 – pagina 83					
6	Ricerca oggetto avanzata	6 – pagina 84					
7	Ricerca completa sul DB	7 – pagina 85					
8	Ricerca oggetto parziale o per carattere	8 – pagina 86					
9	Ricerca bibliografica per autore	9 – pagina 87					
10	Ricerca bibliografica per nome oggetto	10 – pagina 88					
11	Creazione nuovo oggetto da inserire nel DB	11 – pagina 89					
12	Aggiornamenti parametri relativi ad un oggetto	12 – pagina 90					
13	Cancellazione di un oggetto dal DB	13 – pagina 91					
14	Creazione nuovo attributo di un oggetto esistente	14 – pagina 92					
15	Cancellazione attributo di un oggetto	15 – pagina 93					
16	Aggiunta nuova nota bibliografica per un oggetto	16 – pagina 94					
17	Aggiunta nuova nota semplice per un oggetto	17 – pagina 95					
18	Aggiunta nuova immagine per un oggetto nel DB	18 – pagina 96					
19	Visualizzazione lista utenti registrati	19 – pagina 97					







USE CASE 1		Registrazione				
Goal in Context		L'utente vuole ottenere le credenziali per accedere al sistema.				
Preconditions		Utente non registrato.				
Success End Condition		Registrazione a	vvenuta con successo.			
Failed End Condition		Registrazione nor	n avvenuta con successo.			
Primary Actor			Utente.			
Trigger		L'utente accede	al form di Registrazione.			
DESCRIPTION	Step	Utente	GUI			
	1	Preme il pulsante Register Now.				
	2		Mostra la finestra di registrazione.			
	3	Inserisce le informazioni richieste.				
	4	Clicca sul bottone "Submit".				
	5		Invio richiesta di registrazione.			
	6		Riceve conferma lato server.			
	7		Visualizzazione finestra di conferma dell'invio dei dati.			
	8		Manda una e-mail di conferma dell'avvenuta Registrazione.			
EXTENSIONS	Step Utente		GUI			
Utente giá registrato	6a		Riceve un errore server-side.			
	7a		Mostra messaggio di errore.			

Tabella 1 - Registrazione.







USE CASE 2	Login					
Goal in Context		L'utente vuole loggarsi al sistema.				
Preconditions		Utente non lo	oggato.			
Success End Condition		Utente logg	gato.			
Failed End Condition		Utente non lo	oggato.			
Primary Actor		Utente				
Trigger		Pressione taste	o Login.			
DESCRIPTION	Step	Utente	GUI			
	1	Inserisce le credenziali di accesso.				
	2	Preme sul tasto LogIn.				
	3		Effettua una richiesta al server con le credenziali.			
	4		Riceve un messaggio di OK dal server.			
	5		Mostra home page.			
EXTENSIONS	Step	Utente	GUI			
Credenziali non valide	4a		Riceve un errore server-side.			
	5a		Mostra messaggio di errore.			

Tabella 2 - Login.







USE CASE 3	Ricerca per nome				
Goal in Context	L'utente vuole cercare un oggetto inserendo il nome completo.				
Preconditions		Utente la	oggato.		
Success End Condition		Ricerca effettuat	a con successo.		
Failed End Condition		Ricerca effettuata	con insuccesso.		
Primary Actor		Uten	ite.		
Trigger		Pressione tasto Search nel	pannello "Object Name".		
DESCRIPTION	Step	Utente	GUI		
	1	Inserisce il nome dell'oggetto.			
	2	Sceglie il tipo di oggetto.			
	3	Preme il tasto Search.			
	4		Effettua una richiesta al server.		
	5		Riceve un messaggio di OK dal server.		
	6		Mostra risultati nella tab.		
EXTENSIONS	Step	Utente	GUI		
Oggetto non esistente	5a		Riceve un errore server-side.		
	6a		Mostra messaggio di errore.		

Tabella 3 - Ricerca per nome.







USE CASE 4	Ricerca per parametro					
Goal in Context	L'utente vuole cercare un oggetto inserendo il valore di un parametro per un oggetto.					
Preconditions		Utente logg	ato.			
Success End Condition		Ricerca effettuata c	on successo.			
Failed End Condition		Ricerca effettuata co	n insuccesso.			
Primary Actor		Utente.				
Trigger		Pressione tasto Search nel pann	ello "Object Parameter".			
DESCRIPTION	Step	Utente	GUI			
	1	Sceglie tipo di parametro.				
	2	Sceglie tipo di valore.				
	3	Inserisce valore numerico.				
	4	Preme il pulsante Search.				
	5		Effettua una richiesta al server.			
	6		Riceve un messaggio di OK dal server.			
	7		Mostra risultati.			
EXTENSIONS	Step	Utente	GUI			
Parametro non valido	6a		Riceve un errore server-side.			
	7a		Mostra messaggio di errore.			

Tabella 4 - Ricerca per parametro.







USE CASE 5	Ricerca per categoria				
Goal in Context		L'utente vuole cercare un oggetto) inserendo una categoria.		
Preconditions		Utente logg	ato.		
Success End Condition	Ricerca effettuata con successo.				
Failed End Condition		Ricerca effettuata co	n insuccesso.		
Primary Actor	Utente.				
Trigger	Pressione tasto Search nel pannello "Object Category".				
DESCRIPTION	Step	Utente	GUI		
	1	Sceglie tipo di categoria.			
	2 Preme sul tasto Search.				
	3		Mostra Tabella e avviso.		

Tabella 5 - Ricerca per categoria.







USE CASE 6	Ricerca avanzata				
Goal in Context		L'utente vuole visualizzare il p	annello di ricerca avanzata.		
Preconditions		Utente lo	ggato.		
Success End Condition	Visualizzazione effettuata con successo.				
Failed End Condition		Visualizzazione effettu	ata con insuccesso.		
Primary Actor		te.			
Trigger		Pressione tasto Advanced nel p	annello "Advanced Search".		
DESCRIPTION	Step	Utente	GUI		
	1 Sceglie tipo di categoria.				
	2 Preme sul tasto Advanced.				
	3		Mostra Tabella e avviso.		

Tabella 6 - Ricerca avanzata.







USE CASE 7	Ricerca completa			
Goal in Context	L'utente vuole visualizzare l'elenco completo di oggetti.			
Preconditions		Ute	nte loggato.	
Success End Condition		Ricerca effe	ttuata con successo.	
Failed End Condition				
Primary Actor			Utente	
Trigger		Pressione tasto Show all	nel pannello "Advanced Search".	
DESCRIPTION	Step	Utente	GUI	
	1	Pressione tasto Show all.		
	2		Effettua una richiesta al server.	
	3		Riceve un messaggio di OK dal server.	
	4		Mostra risultati.	

Tabella 7 - Ricerca completa.







USE CASE 8	Ricerca per carattere					
Goal in Context		L'utente vuole cercare un oggetto inserendo un carattere.				
Preconditions		Utente logg	ato.			
Success End Condition		Ricerca effettuata c	on successo.			
Failed End Condition		Ricerca effettuata co	n insuccesso.			
Primary Actor		Utente.				
Trigger		Pressione tasto search del pannel	lo "Generic/Partial Chars".			
DESCRIPTION	Step	Utente	GUI			
	1	Inserire carattere.				
	2	Pressione tasto search.				
	3		Effettua una richiesta al server.			
	4		Riceve un messaggio di OK dal server.			
	5		Mostra risultati.			
EXTENSIONS	Step	Utente	GUI			
Parametro non valido	5a		Riceve un errore server-side.			
	6a		Mostra messaggio di errore.			

Tabella 8 - Ricerca per carattere.







USE CASE 9	Ricerca bibliografica per autore					
Goal in Context		L'utente vuole cercare un oggetto i	nserendo il nome di un autore.			
Preconditions		Utente log	ggato.			
Success End Condition		Ricerca effettuata con successo.				
Failed End Condition						
Primary Actor		Utente	2.			
Trigger		Pressione tasto BybliobyAuthor sul pa	annello "Biblio Notes by Author".			
DESCRIPTION	Step	Utente	GUI			
	1	Inserire stringa.				
	2	Pressione tasto BybliobyAuthor.				
	3		Mostra tabella e avviso.			

Tabella 9 - Ricerca bibliografica per autore.







USE CASE 10	Ricerca bibliografica per nome oggetto					
Goal in Context		L'utente vuole cercare un oggetto ins	erendo il nome di un oggetto			
Preconditions		Utente logg	ato			
Success End Condition		Ricerca effettuata c	on successo			
Failed End Condition						
Primary Actor	Utente					
Trigger	Pressione tasto BybliobyObject sul pannello "Biblio Notes by Object".					
DESCRIPTION	Step	Utente	GUI			
	1	Inserire stringa.				
	2	Pressione tasto BybliobyObject				
	3		Mostra Tabella e avviso.			

Tabella 10 - Ricerca bibliografica per nome oggetto.







USE CASE 11	Creazione nuovo oggetto.			
Goal in Context		L'utente vuole creare un nuovo oggetto.		
Preconditions		Utente logga	ato.	
Success End Condition		Oggetto creato con successo.		
Failed End Condition		Oggetto non creato con successo.		
Primary Actor		Utente.		
Trigger		Pressione tasto Ne	w Object.	
DESCRIPTION	Step	Utente	GUI	
	1	Sceglie tipo di oggetto.		
	2	Inserisce i parametri.		
	3	Preme tasto NewVObject.		
	4		Fa una richiesta al server.	
	5		Riceve un messaggio di OK dal server.	
	6		Mostra avviso di operazione effettuata con successo.	
EXTENSIONS	Step	Utente	GUI	
Parametro non valido	5a		Riceve un errore server-side.	
	5a		Mostra messaggio di errore.	

Tabella 11 - Creazione nuovo oggetto.







USE CASE 12	Aggiornamento parametri.			
Goal in Context	L'utente vuole aggiornare i parametri di un attributo.			
Preconditions		Utente logg	ato.	
Success End Condition		Aggiornamento effettuato con successo.		
Failed End Condition		Aggiornamento effettuat	o con insuccesso.	
Primary Actor	Utente.			
Trigger	Pressione tasto Update Object.			
DESCRIPTION	Step	Utente	GUI	
	1	Compila campi dei parametri.		
	2	Preme tasto Update Value.		
	3		Esegue una richiesta al server.	
	4		Riceve un messaggio di OK dal server.	
	5		Mostra avviso di operazione effettuata con successo.	
EXTENSIONS	Step	Utente	GUI	
Parametro non valido	4a		Riceve un errore server-side.	
	5a		Mostra messaggio di errore.	

Tabella 12 - Aggiornamenti parametri.







USE CASE 13	Cancellazione Oggetto.			
Goal in Context		L'utente vuole Cancellare un oggetto.		
Preconditions		Utente lo _{	ggato.	
Success End Condition		Cancellazione effettuata con successo.		
Failed End Condition	Aggiornamento effettuato con insuccesso.			
Primary Actor		Utente.		
Trigger		Pressione tasto Delete Object.		
DESCRIPTION	Step	Utente	GUI	
	1	Compila campi dei parametri.		
	2	Preme tasto Delete Object.		
	3		Compie una richiesta al server.	
	4		Riceve un messaggio di OK dal server.	
	5		Mostra avviso di operazione effettuata con successo.	
EXTENSIONS	Step	Utente	GUI	
Parametro non valido	4a		Riceve un errore server-side.	
	5a		Mostra messaggio di errore.	

Tabella 13 - Cancellazione Oggetto.







USE CASE 14	Creazione nuovo attributo.		
Goal in Context	L'utente vuole creare un nuovo attributo.		
Preconditions		Utente logg	ato.
Success End Condition		Creazione effettuata	con successo.
Failed End Condition		Creazione effettuata c	on insuccesso.
Primary Actor	Utente.		
Trigger	Pressione tasto New Attribute.		
DESCRIPTION	Step	Utente	GUI
	1	Compila campi del nuovo attributo.	
	2	Preme tasto New Attribute.	
	3		Compie una richiesta al server.
	4		Riceve un messaggio di OK dal server.
	5		Mostra avviso di operazione effettuata con successo.
EXTENSIONS	Step	Utente	GUI
Parametro non valido	4a		Riceve un errore server-side.
	5a		Mostra messaggio di errore.

Tabella 14 - Creazione nuovo attributo.







USE CASE 15	Cancellazione nuovo attributo.		
Goal in Context	L'utente vuole cancellare un nuovo attributo.		
Preconditions		Utente log	ggato.
Success End Condition	Cancellazione effettuata con successo.		
Failed End Condition	Cancellazione effettuata con insuccesso.		
Primary Actor	Utente.		
Trigger		Pressione tasto De	elete Attribute.
DESCRIPTION	Step	Utente	GUI
	1	Compila campi per la cancellazione attributo.	
	2	Preme tasto Delete Attribute.	
	3		Compie una richiesta al server.
	4		Riceve un messaggio di OK dal server.
	5		Mostra avviso di operazione effettuata con successo.
EXTENSIONS	Step	Utente	GUI
Parametro non valido	4a		Riceve un errore server-side.
	5a		Mostra messaggio di errore.

Tabella 15 - Cancellazione nuovo attributo.







USE CASE 16	Aggiunta nuova nota bibliografica.				
Goal in Context		L'utente vuole inserire una nuova nota bibliografica.			
Preconditions		Utente log	gato.		
Success End Condition		Creazione effettuata con successo.			
Failed End Condition	Creazione effettuata con insuccesso.				
Primary Actor		Utente.			
Trigger		Pressione tasto	New Biblio.		
DESCRIPTION	Step	Utente	GUI		
	1	Compila campi per la creazione nota bibliografica.			
	2	Preme tasto New Biblio.			
	3		Compie una richiesta al server.		
	4		Riceve un messaggio di OK dal server.		
	5		Mostra avviso di operazione effettuata con successo.		
EXTENSIONS	Step	Utente	GUI		
Parametro non valido	4a		Riceve un errore server-side.		
	5a		Mostra messaggio di errore.		

Tabella 16 - Aggiunta nuova nota bibliografica.







USE CASE 17	Aggiunta nuova nota semplice.			
Goal in Context	L'utente vuole inserire una nuova nota semplice.			
Preconditions		Utente logg	ato.	
Success End Condition		Creazione effettuata con successo.		
Failed End Condition		Creazione effettuata con insuccesso.		
Primary Actor	Utente.			
Trigger	Pressione tasto New Note.			
DESCRIPTION	Step	Utente	GUI	
	1	Compila campi per la creazione nota semplice.		
	2	Preme tasto New Note.		
	3		Compie una richiesta al server.	
	4		Riceve un messaggio di OK dal server.	
	5		Mostra avviso di operazione effettuata con successo.	
EXTENSIONS	Step	Utente	GUI	
Parametro non valido	4a		Riceve un errore server-side.	
	5a		Mostra messaggio di errore.	

Tabella 17 - Aggiunta nuova nota semplice.







USE CASE 18	Aggiunta nuova immagine		
Goal in Context		L'utente vuole inserire una	a nuova Immagine.
Preconditions		Utente logg	ato.
Success End Condition	Inserimento effettuato con successo.		
Failed End Condition			
Primary Actor	Utente.		
Trigger	Pressione tasto New Image.		
DESCRIPTION	Step	Step Utente GUI	
	1	Seleziona la sorgente.	
	2	Inserisce la path.	
	3	Preme tasto upload.	
	4		Mostra avviso.

Tabella 18 - Aggiunta nuova immagine.







USE CASE 19	Visualizzazione lista utenti.		
Goal in Context	L'amministratore vuole visualizzare la lista utenti.		
Preconditions	Admin loggato.		
Success End Condition	Visualizzazione effettuata.		
Failed End Condition	Visualizzazione non effettuata.		
Primary Actor	Admin.		
Trigger	Pressione tasto User list.		
DESCRIPTION	Step	Utente	GUI
	1		Mostra griglia con lista utenti.
EXTENSIONS	Step Utente GUI		
Login come utente normale	1a		Pannello di avviso per permessi non sufficienti.

Tabella 19 - Visualizzazione lista utenti.







9. Ringraziamenti

Al Project Manager del gruppo DAME Dottor Massimo Brescia, ricercatore astronomo presso l'INAF OAC Napoli per avermi dato la possibilità di svolgere il tirocinio presso l'Osservatorio Astronomico di Capodimonte e per la costante disponibilità e cortesia avute nei miei confronti.

Al professor Giuseppe Longo ordinario di Astrofisica della Federico II e Principal Investigator del gruppo DAME.

Al Dottor Marco Castellani, astronomo ricercatore dell'INAF presso l'Osservatorio di Roma in qualità di consulente scientifico per i Globular Clusters.

Al professor Guido Russo per i consigli ed i suggerimenti profusi durante la stesura della tesi.

Ringrazio infine tutti i miei amici e i collaboratori del gruppo DAME che mi hanno aiutato e sostenuto nel mio lavoro.



10. Bibliografia

1. IVOA. [Online] http://www.ivoa.net/.

VERSITA²DEGU STUDI DI

- 2. S.Co.P.E. [Online] http://www.scope.unina.it/C19/astrophysics-gridcomputing/default.aspx.
- 3. AJAX. [Online] http://www.dmoz.org/Computers/Programming/Languages/JavaScript/Ajax/.
- 4. GWT. [Online] http://code.google.com/intl/it-IT/webtoolkit/.
- 5. SMARTGWT. [Online] http://code.google.com/p/smartgwt/.
- 6. DAME. [Online] http://dame.dsf.unina.it/.
- 7. Machine learning. [Online] http://dame.dsf.unina.it/machine_learning.html.
- 8. VO-compliant. [Online] http://www.euro-vo.org/pub/dca/tutorials.html.
- 9. Harris 2003. [Online] http://physwww.physics.mcmaster.ca/~harris/mwgc_2003.dat.
- 10. M31. [Online] http://iopscience.iop.org/1538-3881/133/6/2764/205732.text.html.

11. **Clement, C. M., Hilditch, R. W., Kaluzny, J. and Rucinski.** Evolution of Horizontal Branch Stars in Globular Clusters: The Interesting Case of V79 in M3. *Astrophysical Journal Letters*. 1997.

- 12. DAMEWARE. [Online] http://dame.dsf.unina.it/beta_info.html.
- 13. Castellani, Vittorio. Fondamenti di Astrofisica Stellare. s.l. : Zanichelli, 1985.
- 14. Generazioni di stelle. [Online] http://it.wikipedia.org/wiki/Popolazioni_stellari.
- 15. Il catalogo di Messier. [Online] http://messier.seds.org/.
- 16. New General Catalogue. [Online] http://spider.seds.org/ngc/ngc.html.
- 17. Pulsar. [Online] http://pulsar.ca.astro.it/pulsar/IT_PSR.html.
- 18. Stella. [Online] http://it.wikipedia.org/wiki/Stella.
- 19. Horizontal branch star. [Online] http://en.wikipedia.org/wiki/Horizontal_branch.
- 20. Diagramma HR. [Online] http://it.wikipedia.org/wiki/Diagramma_Hertzsprung-Russell.
- 21. Text Mining. [Online] http://en.wikipedia.org/wiki/Text_Mining.
- 22. Data Mining. [Online] http://it.wikipedia.org/wiki/Data_Mining.
- 23. VO Standard. [Online] http://www.us-vo.org/oldhome/summer-school/2004/proceedings/presentations/SummerSchoolClientSide.htm#VO_Standard.





24. Linee guida di partecipazione IVOA. [Online] http://www.ivoa.net/Documents/latest/IVOAParticipation.html.

25. RIA. [Online] http://www2.mokabyte.it/cms/article.run?articleId=THB-TYI-QI9-780_7f000001_17548745_899249ca.

- 26. XML. [Online] http://www.w3.org/XML/.
- 27. CSS. [Online] http://www.w3.org/Style/CSS/.
- 28. DOM. [Online] http://www.w3.org/DOM/.
- 29. javascript. [Online] http://www.w3schools.com/js/.
- 30. GWT SDK. [Online] http://code.google.com/intl/it-IT/webtoolkit/download.html.

31. JRE emulation library. [Online] http://code.google.com/p/google-web-toolkit-doc-1-5/wiki/ReleaseNotes_1_5_JreEmulation.

32. Java Serialization API. [Online] http://java.sun.com/developer/technicalArticles/Programming/serialization/.

- 33. GWT RPC. [Online] http://code.google.com/intl/it-IT/webtoolkit/doc/latest/tutorial/RPC.html.
- 34. Isomorphic Software. [Online] http://www.smartclient.com/.
- 35. LGPL. [Online] http://www.gnu.org/licenses/lgpl.html.
- 36. Lista Librerie Widget. [Online] http://en.wikipedia.org/wiki/List_of_widget_toolkits.
- 37. Datasource. [Online]

http://www.smartclient.com/smartgwt/javadoc/com/smartgwt/client/data/DataSource.html.

- 38. RSS. [Online] http://feed2.w3.org/docs/rss2.html.
- 39. STILTS. [Online] http://www.star.bris.ac.uk/~mbt/stilts/.
- 40. Alistair Cockburn. [Online] http://alistair.cockburn.us/.