

**UNIVERSITA' DEGLI STUDI DI PADOVA**



**FACOLTA' DI SCIENZE STATISTICHE**

**CORSO DI LAUREA IN STATISTICA E TECNOLOGIE INFORMATICHE**

**TESI DI LAUREA TRIENNALE**

**REALIZZAZIONE DI UN'APPLICAZIONE WEB PER LA GESTIONE  
DEI CESPITI IN UN CONTESTO MULTINAZIONALE**

**Relatore: Prof.ssa SUSI DULLI**

**Correlatore: Dott.ssa WIEBKE CORSI**

**Laureando: ANDREA SALIN**

**ANNO ACCADEMICO 2004 – 2005**



---

**UNIVERSITA' DEGLI STUDI DI PADOVA**



**Facoltà di Scienze Statistiche**

**Corso di Laurea in Statistica e Tecnologie Informatiche**

**TESI DI LAUREA TRIENNALE**

**REALIZZAZIONE DI UN'APPLICAZIONE WEB  
PER LA GESTIONE DEI CESPITI  
IN UN CONTESTO MULTINAZIONALE**

Relatore: Prof.ssa SUSI DULLI

Correlatore: Dott.ssa WIEBKE CORSI

**Laureando: ANDREA SALIN**

**ANNO ACCADEMICO 2004 – 2005**



---

Ringraziamenti:

- Wiebke Corsi, Enrico Liboni, Michele Franzo, Giacomo Ferrari e tutto l'ambiente di lavoro Alcoa IT, per la disponibilità dimostrata nei miei confronti
- L'azienda Alcoa, per avermi dato la possibilità di svolgere lo stage
- Il mio relatore, Prof.ssa Susi Dulli, per i preziosi consigli e suggerimenti
- La mia famiglia, per avermi sempre sostenuto durante i miei anni all'università
- Tutti i miei amici e i compagni di università, per tutti i bei momenti passati insieme durante questi anni
- Giorgia, per la gioia che ogni giorno mi dona il suo sorriso

*Andrea*



## **Indice**

<b>Introduzione</b>	<b>3</b>
<b>Capitolo 1</b>	<b>5</b>
<b>Il contesto aziendale</b>	<b>5</b>
<b>1.1 Alcoa</b>	<b>5</b>
1.1.1 La storia di Alcoa	6
1.1.2 I segmenti di produzione e il mercato	7
1.1.3 Alcoa in Europa	10
<b>1.2 La struttura organizzativa</b>	<b>11</b>
<b>1.3 Alcoa Business System</b>	<b>12</b>
<b>1.4 Il sistema informativo</b>	<b>13</b>
1.4.1 L'architettura della rete	14
1.4.2 Il sistema ERP: le Oracle Applications	15
<b>1.5 Web &amp; Reporting Center (WRC)</b>	<b>16</b>
<b>Capitolo 2</b>	<b>17</b>
<b>Definizione ed analisi del progetto</b>	<b>17</b>
<b>2.1 L'analisi del problema</b>	<b>17</b>
2.1.1 La situazione attuale	18
<b>2.2 La nuova soluzione</b>	<b>19</b>
2.2.1 I codici a barre	21
2.2.2 Finalità e Vantaggi	23
<b>2.3 Le scelte tecnologiche</b>	<b>24</b>
2.3.1 Architettura di un'applicazione web	24
2.3.2 Servlet, JSP e Tomcat	26
2.3.3 RDBMS Oracle	29

<b>Capitolo 3</b>	<b>31</b>
<b>La realizzazione del progetto</b>	<b>31</b>
<b>3.1 Definizione dei requisiti funzionali</b>	<b>31</b>
3.1.1 Autenticazione utenti e privilegi	31
3.1.2 Operazioni eseguibili	33
<b>3.2 Il Database</b>	<b>34</b>
3.2.1 La progettazione del database	35
<b>3.3 L'implementazione e il layout</b>	<b>39</b>
3.3.1 Autenticazione	39
3.3.2 Review DATA	43
3.3.3 Upload DATA	46
3.3.4 View DATA	48
<b>3.4 Il deployment</b>	<b>50</b>
3.4.1 La migrazione	50
<b>Conclusioni</b>	<b>51</b>
<b>Bibliografia</b>	<b>53</b>

## **Introduzione**

Oggi, qualsiasi realtà aziendale ha la necessità di tenere sotto controllo l'evolversi delle proprie attività facendo in modo di renderle il più efficienti ed efficaci possibile.

Tale supervisione permette ad un'azienda di trovare le soluzioni più vantaggiose in termini di *performance* e di costi di gestione per qualsiasi tipo di attività.

La gestione dei beni che sono elementi attivi del patrimonio di un'azienda, detti anche cespiti, rappresenta una delle principali problematiche di cui tener traccia, soprattutto nel caso di aziende di grandi dimensioni.

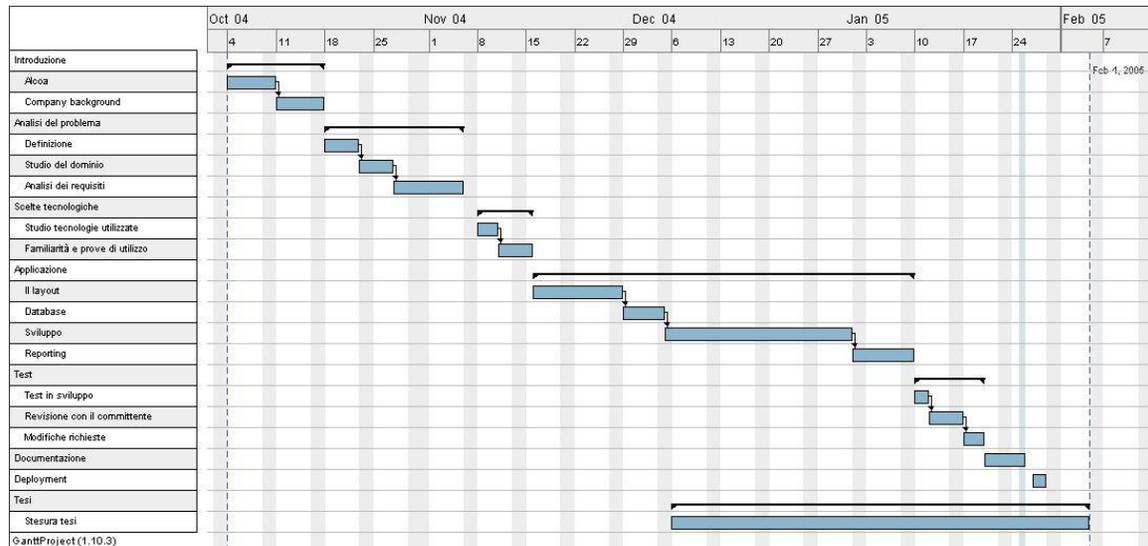
Questa tesi ha come obiettivo la progettazione e l'implementazione di un sistema chiamato *Asset Application*, che permette la gestione automatizzata dei cespiti e del loro inventario. Il progetto è stato realizzato durante un periodo di stage, presso l'Information Technology di Alcoa, azienda multinazionale leader mondiale nella produzione di alluminio primario, alluminio costruito e allumina. Più precisamente all'interno di un team europeo, il *Web & Reporting Center (WRC)*, con sede a Marcon (Venezia).

Il WRC ha il compito di gestire tutti gli aspetti legati al web della rete intranet e al reporting di dati. Fornisce applicazioni e servizi, costruiti su richiesta, per particolari segmenti produttivi o business unit.

*Asset Application* è un *tool* condiviso che permette agli utenti registrati di eseguire alcune operazioni sui dati relativi ai cespiti presenti in ogni *location* europea. Sono possibili l'aggiornamento di alcune informazioni relative ad uno specifico cespite, la loro visualizzazione a livello web, con la possibilità di esportare i dati in formato *excel*, e l'aggiornamento dell'inventario fisico dei cespiti tramite uno sheet excel specifico e realizzato appositamente per interfacciarsi con il tool.

Il sistema è stato progettato e realizzato per semplificare e rendere più sicuro questo trasferimento di dati, automatizzandolo, e, quindi, facendolo diventare più veloce e riducendo la probabilità di errori.

Di seguito viene riportato il diagramma di *Gantt* che descrive le varie fasi di sviluppo del progetto realizzato durante l'attività di stage.



**Figura I.1 Diagramma di Gantt**

Gli argomenti introdotti vengono trattati nello svolgersi dei seguenti capitoli:

### **Capitolo 1: Il contesto aziendale**

Il primo capitolo fornisce una visione complessiva dell'azienda, accennando alla sua storia, alla struttura organizzativa e ai mercati in cui opera. Viene anche descritto a grandi linee il sistema informativo presente in azienda per capire come Alcoa gestisce le proprie risorse informatiche.

### **Capitolo 2: Definizione ed analisi del progetto**

Nel secondo capitolo si descrive il progetto realizzato, analizzando vecchia e nuova soluzione con relativi svantaggi e vantaggi. Inoltre vengono evidenziate le scelte tecnologiche adottate come strumento di realizzazione.

### **Capitolo 3: La realizzazione del progetto**

Il terzo capitolo espone il progetto nel dettaglio, descrivendo i requisiti funzionali e le varie fasi di progettazione. Si analizzano la costruzione del database e l'implementazione delle pagine web accennando, infine, al processo di *deployment* dell'applicazione.

## Capitolo 1

### Il contesto aziendale

---

#### 1.1 Alcoa

Alcoa è una multinazionale leader mondiale nella produzione e nella prima lavorazione dell' alluminio. Il nome dell'azienda era l'acronimo di *Aluminium Company of America* ma in realtà si tratta di una compagnia che basa il proprio business non solo sull'alluminio, ma è attiva anche in tutti i principali settori industriali che hanno legami con esso: da quello aerospaziale a quello automobilistico, dal chimico all'edilizia, dall'imballaggio allo sport.

E' un'azienda fondata sul lavoro, sulla condivisione globale di tecniche, esperienze e business, e mira ad essere la scelta migliore per i propri clienti. La sensazione che Alcoa vuole dare è di collaborazione, per rendere stabile e duratura la spinta produttiva di un cliente, e per crescere assieme nei vari business.

### 1.1.1 La storia di Alcoa

Nel 1888 il chimico Charles Martin Hall fondò la Pittsburgh Reduction Company, in un periodo in cui l'alluminio era un metallo semi-prezioso, addirittura più raro dell'argento. Essa fu la prima a produrre e commercializzare oggetti in alluminio utilizzando la tecnica sviluppata dal suo fondatore per la trasformazione dalla bauxite in ossido di alluminio – allumina. In pochi anni il prezzo dell'alluminio decrementò fortemente, facilitando un aumento sostanziale dei business. Le prime produzioni comprendevano utensili da cucina, cavi elettrici, carrozzerie di automobili; da citare anche l'utilizzo da parte dei fratelli Wright di un motore costituito da parti in alluminio per intraprendere il primo volo della storia.

Nel 1907 l'organizzazione si era estesa in tutto il territorio degli Stati Uniti e in Canada, tanto che venne cambiato il nome in *Aluminium Company of America*. Successivamente, con l'espansione a livello globale, venne esteso ad Alcoa Inc.

Durante la seconda guerra mondiale la richiesta di alluminio aumentò radicalmente fino a raddoppiare il proprio volume, consentendo ad Alcoa di proseguire efficacemente nel suo progressivo sviluppo.

Negli ultimi anni Alcoa ha significativamente incrementato la propria presenza nel mondo grazie alla continua crescita aziendale, alle alleanze globali e alle grandi acquisizioni sia in Europa che negli Stati Uniti.

Con la crescita della domanda cresceva di conseguenza la competitività nei mercati. Per rispondere a tale agguerrita concorrenza internazionale, Alcoa decise di allargare la propria base tecnologica, perfezionando i processi produttivi, abbattendo i costi, ingrandendo le linee produttive e i propri mercati.

L'alluminio, con le sue caratteristiche di robustezza, leggerezza, riciclabilità, stava diventando la scelta ottimale per migliaia di prodotti vecchi e nuovi.

Molte cose sono cambiate, ma la certezza rimane una sola: dal primo giorno ad oggi Alcoa è rimasta leader a livello mondiale nella produzione e prima lavorazione dell'alluminio.

Oggi l'azienda conta 120.000 dipendenti in 41 differenti paesi, più di 350 locazioni in tutto il mondo (Figura 1.1), e un fatturato nel 2003 di 21.5 miliardi di dollari.

**Map: Alcoa Around the World**



**Figura 1.1 Il mondo Alcoa**

### 1.1.2 I segmenti di produzione e il mercato

L'attività produttiva di Alcoa può essere identificata da cinque diverse fasi:

- Allumina e prodotti chimici (Alumina and Chemicals)  
Comprende l'estrazione della bauxite e il processo di trasformazione in allumina.
- Settore primario dei metalli (Primary Metals)  
Raggruppa tutte le fonderie di Alcoa che producono alluminio grezzo e particolari leghe metalliche che vengono in parte vendute a clienti esterni ma, soprattutto, vengono reintrodotti in Alcoa e trasferite agli altri segmenti di produzione.

- **Prodotti laminati (Flat-Rolled Products)**  
Si occupa della produzione e della vendita di alluminio sotto forma di placche, fogli e lamine, ma anche di lattine utilizzate per contenere bevande e liquidi in generale, nonché di prodotti laminati per il trasporto, l'edilizia e la distribuzione.
  - **Prodotti ingegneristici (Engineered Products)**  
Riguarda gli estrusi in lega e le forgiature, ovvero le manifatture di alluminio o di una sua lega lavorata per poter essere utilizzata direttamente nell'assemblaggio di oggetti finiti. Sono prodotti dei quali usufruiscono il settore del trasporto e dell'edilizia.
  - **Imballaggio e prodotti di consumo (Packaging and Consumer)**  
Include gli imballi flessibili, prodotti di consumo, sistemi di chiusura per cibi e bevande, bottiglie in PET (polyethylene terephthalate) e macchine per l'imballaggio.
- Tutto ciò che non riguarda uno di questi cinque segmenti viene classificato sotto la voce altro (Other). Questo gruppo comprende AFL, che produce componenti elettrici, cavi in fibra ottica e fornisce servizi all'industria delle telecomunicazioni, delle costruzioni e dell'automobile.

Il grafico mostra la ripartizione del fatturato annuale del 2003 divisa per segmenti di produzione (Figura 1.2).



**Figura 1.2 Fatturato per settore**

I segmenti produttivi descritti entrano in gioco in vari settori del business Alcoa. I mercati principali con cui la multinazionale ha a che fare sono:

### Mercato Aerospaziale (*Aerospace*)

Alcoa è protagonista nel mercato aerospaziale per applicazioni civili e militari. I prodotti Alcoa sono ampiamente utilizzati per la realizzazione dei motori degli aeroplani, per sistemi di fissaggio, leghe avanzate per la fusoliera, per le ali, per gli ingranaggi e per le ruote di sbarco

### Mercato Automobilistico (*Automotive*)

L'importante sviluppo del settore automobilistico negli ultimi decenni ha portato questo mercato ad essere uno dei più redditizi per Alcoa, i cui prodotti vanno a comporre diverse parti dell'insieme di un veicolo; dalle sospensioni alla carrozzeria, fino ad arrivare alle ruote ed ad altri sistemi elettrici di distribuzione.

### Mercato dell'Imballaggio e dei prodotti di consumo (*Packaging and Consumer*)

Questo mercato è molto diversificato e non comprende solo prodotti derivati dall'alluminio, ma anche da materiali plastici. Quindi, oltre alla produzione di lattine in alluminio (Coca-Cola Company è tra i clienti Alcoa), produce anche bottiglie in PVC e svariati tipi di sistemi di chiusura.

### Mercato della Distribuzione (*Commercial Transportation*)

Le entrate da questo segmento di mercato derivano principalmente dalla vendita diretta a terzi di estrusi, laminati e placche di alluminio, per quelle aziende che intendono svolgere ulteriori lavorazioni sul prodotto.

### Mercato Edilizio (*Building and Construction*)

Il settore delle costruzioni fa largo uso di materiali innovativi tra cui l'alluminio, per la realizzazione di finiture interne e altri accessori.

### Mercato dell'Alluminio e Allumina (*Aluminium and Alumina*)

Alcoa è in assoluto il primo produttore al mondo di allumina, la sostanza dalla quale si produce l'alluminio. Per questo, nonostante il grande fabbisogno interno, le vendite verso altre aziende coprono una buona fetta del mercato.

Mercato dei Prodotti Industriali e Altro (*Industrial Products and Other*)

Vengono incluse vendite di determinati prodotti e servizi per l'industria delle telecomunicazioni, prodotti chimici, attrezzature per lo sport e una vasta gamma di applicazioni industriali.

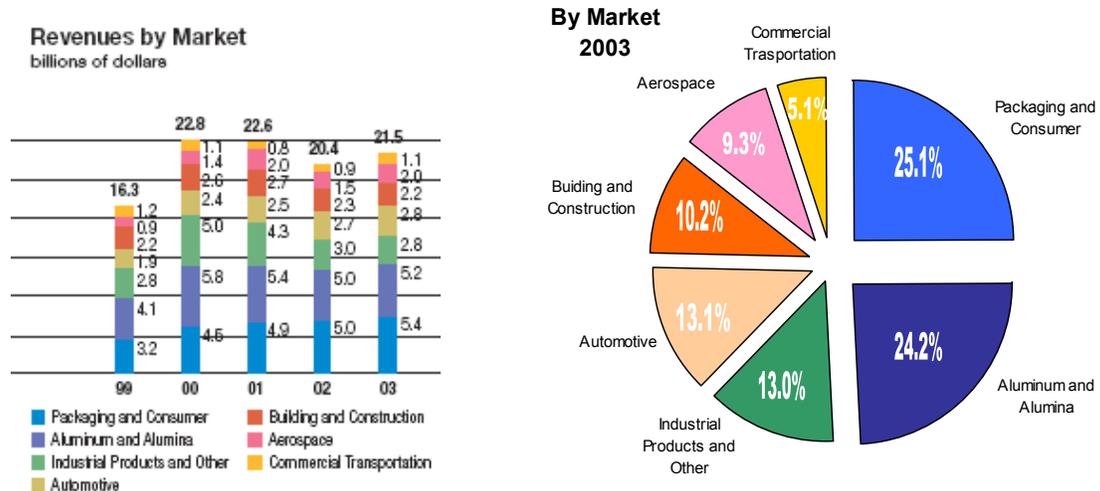


Figura 1.3 Fatturato di Alcoa per settore (1999-2003) e ripartizione del mercato (2003)

1.1.3 Alcoa in Europa

In Europa il gruppo Alcoa si è fortemente sviluppato grazie alla politica d'acquisizione condotta. Il processo di espansione comincia a rilento nel 1965 con la creazione di *Alcoa Netherlands*, ma la vera crescita aziendale nel vecchio continente si verifica nell'ultimo decennio, a cavallo tra il ventesimo e il ventunesimo secolo. Nel 1998 vengono acquisite Alumax e Inespal e nel 2000 la multinazionale Reynolds Metal Company che premise un notevole allargamento del gruppo passando da 68 a 75 unità operative. Nel 2001 viene conglobata AFL (Siemens) per avere un aggancio con il mercato delle telecomunicazioni e nel 2002 viene completata l'acquisizione dei beni della Fairchild Fasteners, una compagnia di supporto per il mercato aerospaziale.

Oggi in Europa Alcoa è presente in ben 16 paesi (Figura 1.4), conta 93 locazioni e approssimativamente 28.000 dipendenti che contribuiscono a un fatturato di più del 20% di quello complessivo dell'azienda.

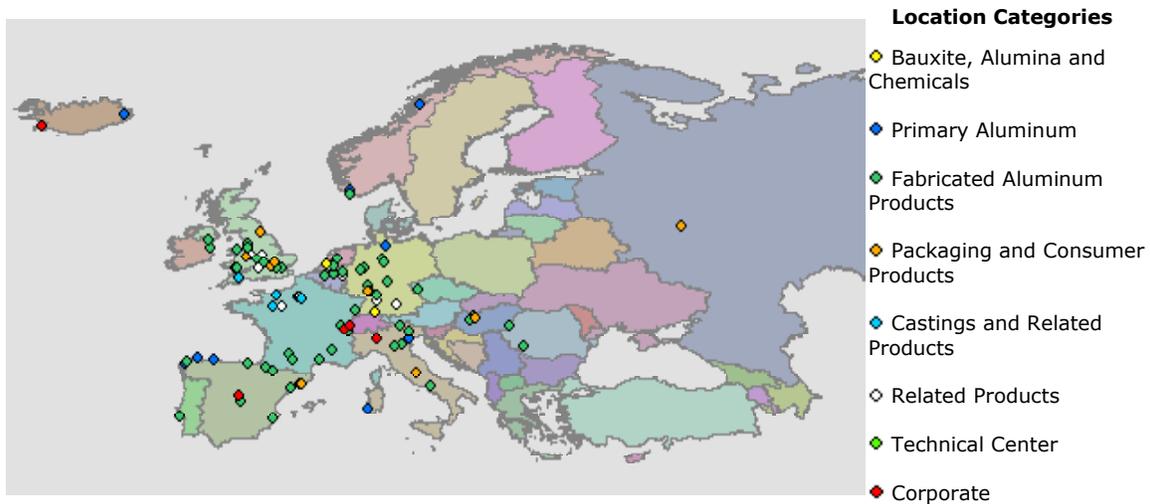


Figura 1.4 Le *locations* di Alcoa in Europa

## 1.2 La struttura organizzativa

Una realtà multinazionale come Alcoa deve posare su di una solida struttura organizzativa che centralizzi in specializzate aree di gestione e controllo tutte le attività che vanno al di là della produzione.

Il quartier generale è sito a Pittsburg (Pennsylvania, USA), ma esistono diverse altre *locations* in cui operano diversi centri direzionali suddivisi per area commerciale e geografica. In particolare l'*Head Quarter* Europeo, dove vengono prese diverse decisioni di carattere strategico e commerciale per gli stabilimenti del vecchio continente, sorge a Ginevra (Svizzera).

Ogni stabilimento necessita della gestione delle infrastrutture informatiche. L'Europa è suddivisa in macro-regioni, l'Italia è una di queste, che devono poter essere gestite in modo centralizzato; risultano quindi indispensabili i servizi forniti dai dipartimenti di

*Information Technology* di ognuna di queste aree geografiche. In Italia tale dipartimento ha sede a Marcon (VE), dove si è svolta l'esperienza di stage.

La struttura organizzativa comprende gli *Shared Service Center* (SSC) che centralizzano e gestiscono tutte le operazioni non direttamente collegate con la produzione. In Europa tale centro servizi ha sede a Székesfehérvár (Ungheria) dove si trovano l'*European Data Center* (EDC), centro nevralgico di tutto l'IT europeo di Alcoa e l'*European Administration Center* (EAC) che gestisce parte della contabilità aziendale per tutti i paesi europei.

Sulla base di una buona struttura organizzativa e per ottenere un controllo completo in tutti i processi aziendali è necessario disporre di un contesto uniforme tanto per le applicazioni che per le infrastrutture. Per questo è stato avviato un progetto di standardizzazione a livello globale chiamato *Enterprise Business Solution* (EBS) con lo scopo di integrare tutte le *Business Unit* in un unico sistema informativo. Si è puntato, quindi, all'unificazione della gestione di tutti i processi aziendali e al controllo capillare di tutte le attività di Alcoa nel mondo.

Su queste basi, nel Luglio 2003, è nata GBS (*Global Business Services*), un'organizzazione che raggruppa tutte le funzioni che forniscono servizi di supporto al Business di Alcoa nel mondo.

### **1.3 Alcoa Business System**

Alcoa Business System è un insieme integrato di principi e applicazioni utilizzati per controllare i business di Alcoa. L'azienda si serve della preparazione, della capacità e del talento delle persone per eliminare gli sprechi e per fornire ai clienti i prodotti e i servizi di cui necessitano. Il dipartimento dell' *Information Technology*, come del resto tutte le altre aree produttive, utilizza ABS per controllare il proprio operato.

Tale sistema si basa su tre concetti fondamentali:

- Produrre per usare

Il punto di partenza di questo principio è rappresentato dal cliente e dalle sue esigenze. Alcoa si pone l'obiettivo di realizzare ciò che il cliente desidera, nei tempi richiesti e nelle quantità desiderate, con il minor costo possibile.

- Eliminare gli sprechi

Si basa sull'individuazione dei problemi, sul dove e sul quando si verificano, per cercare di risolverli e quindi realizzare prodotti il più perfetti possibile. Si cerca sempre di ridurre i costi, migliorare la qualità e la velocità di tutti i processi produttivi e commerciali.

- Le persone come perno del sistema

Il terzo principio è quello che raccoglie un po' tutti i valori dell'ABS. Infatti le persone, sfruttando le proprie capacità professionali nell'interagire con le macchine utensili, con i processi aziendali e con i colleghi, sono le artefici dell'applicazione dei due concetti precedenti. La situazione ottimale a cui si vorrebbe arrivare prevede che le persone allineino la produttività del sistema ai ritmi dei clienti, come suggerisce il primo punto, e operino perfettamente nel sistema migliorandolo incessantemente, come vuole il secondo.

## **1.4 Il sistema informativo**

Le esigenze informative di una multinazionale come Alcoa assumono indubbiamente un'importanza non trascurabile all'interno del contesto aziendale, in quanto, oggi, contribuiscono non solo all'automazione delle procedure operative, ma incidono anche nella definizione e nel perseguimento delle strategie aziendali.

Le diverse aree funzionali devono cercare di perseguire l'obiettivo comune di crescita a livello globale e per poter far questo è indispensabile una facilità di intercomunicabilità fra di esse.

Un efficiente sistema informativo in grado di coordinare le differenti attività aziendali risulta quindi un punto di partenza fondamentale per Alcoa. L'integrazione informatica inoltre rende omogenea ed accessibile la distribuzione delle informazioni all'interno dell'azienda, permettendo la loro condivisione in modo più rapido e semplificando le attività di supporto e manutenzione.

### 1.4.1 L'architettura della rete

La rete di Alcoa è costituita complessivamente di quattro WAN (Wide Area Network) interconnesse fra loro: l'*Australian WAN*, la *South America WAN*, l'*AFL WAN* e la *Corporate WAN*.

Le prime tre sono gestite dalle *Business Unit* che le hanno realizzate mentre la *Corporate WAN* è amministrata da un'apposita organizzazione chiamata *GIS Network Engineering*. Quest'ultima è la più complessa tra le quattro dato che deve gestire molti business e attraversare le giurisdizioni di diversi paesi. La dorsale europea fa parte appunto di questa network che, in principio, era composta da cinque nodi principali: Spagna, Ungheria, Germania, Olanda e Italia. In Europa le connessioni verso l'esterno della rete passavano per il nodo di Kofem/ Székesfehérvár (Ungheria), dotato di un firewall che garantiva una certa sicurezza nella network e forniva un controllo degli accessi alla rete internet.

Il nodo italiano della *Corporate WAN* era situato a Marcon (VE) e rappresentava il gateway utilizzato da tutti gli stabilimenti italiani per avere accesso alla rete aziendale. Questo sistema garantiva i servizi di intercomunicazione come la posta elettronica, l'accesso a internet, la teleconferenza e tutti gli applicativi per la gestione delle attività finanziarie tra i quali il sistema *ERP (Enterprise Resource Planning)* delle *Oracle Applications*.

Questo tipo di architettura, anche se perfettamente funzionante, è stato da poco sostituito, almeno in Europa, da un tipo di configurazione che prevede l'utilizzo del protocollo *MPLS (Multi Protocol Label Switching)* come principale innovazione del sistema.

Questo protocollo è in grado di indirizzare il traffico direttamente verso la destinazione prevista senza dover transitare obbligatoriamente per i nodi principali (nel caso italiano, quello di Marcon). In pratica questa nuova architettura ha reso più snello il traffico d'informazioni, dato che non esistono più punti critici per la rete che devono indirizzare elevate quantità di dati, e, di conseguenza, sono diminuiti i relativi costi di gestione.

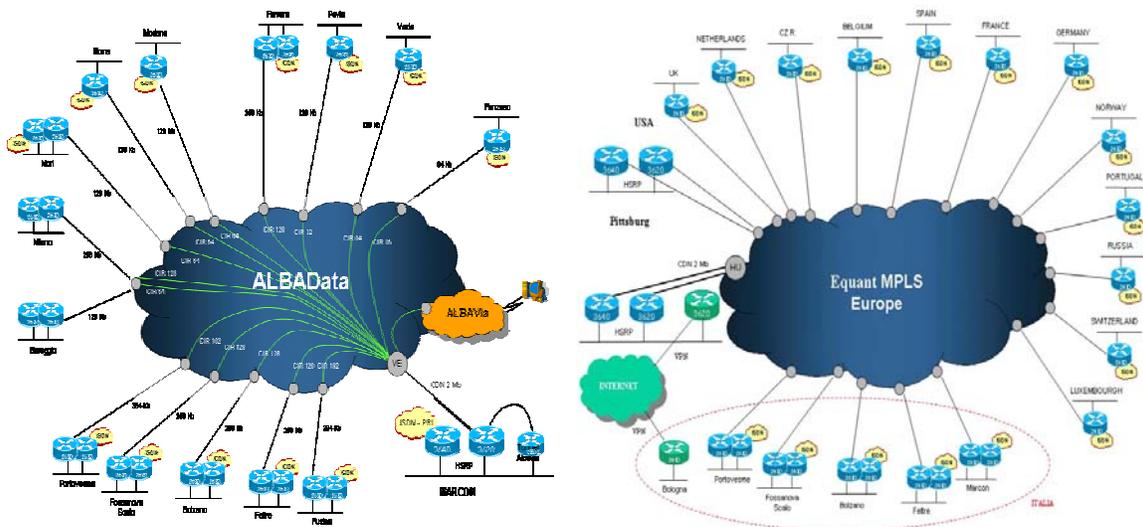


Figura 1.5 Architettura della rete (vecchia e nuova)

### 1.4.2 Il sistema ERP: le Oracle Applications

Un ERP è un gestionale, cioè un pacchetto software formato da vari moduli che si integrano fra loro, la cui caratteristica principale è la gestione integrata di tutte le risorse che partecipano alla creazione dei prodotti/servizi di un'azienda. Tale sistema semplifica e gestisce le operazioni strategiche dell'azienda, quali la gestione dell'inventario, l'interazione con i fornitori, la pianificazione dei prodotti, l'acquisto, l'assistenza ai clienti e la rintracciabilità degli ordini. Supportato da un software applicativo basato su più moduli, collega tra loro operazioni frammentate e condivide i dati attraverso un insieme integrato di moduli applicativi.

Per concretizzare effettivamente il progetto EBS Alcoa ha sottoscritto un accordo con Oracle nel Maggio del 2000 per la fornitura dei database *8i* e del software *Oracle Applications 11i*, il primo sistema ERP a sfruttare la tecnologia e il concetto di *internetworking* offerto da Java, unendo un accattivante aspetto grafico ad una vasta gamma di funzionalità per la gestione aziendale. E' un sistema articolato in più di novanta moduli, ognuno dei quali tratta uno specifico settore delle attività aziendali.

Oracle Applications automatizza, razionalizza e semplifica tutti i processi di business e conserva tutte le informazioni in un unico luogo, in modo che l'azienda possa condividere dati di alta qualità in tempo reale.

Parallelamente esistono altri pacchetti software come EIS e FLITS basati su sistemi AS/400 che riguardano il manufacturing e la fatturazione nei vari settori di business, sviluppati da Alcoa appositamente per venire incontro a particolari esigenze.

### **1.5 Web & Reporting Center (WRC)**

Il Web & Reporting Center è un team europeo che si occupa di tutti gli aspetti legati al web intranet, internet, e-commerce e reporting dei dati. Questo team, creato in Italia all'inizio del 2003, è stato un'estensione di uno già esistente che curava il reporting per l'Europa. Le principali attività di cui si occupa il WRC sono:

- Sviluppo di nuove applicazioni web
- Mantenimento e miglioramento dei siti intranet esistenti
- EBS reporting in collaborazione con i team di supporto funzionali
- Supporto tecnico allo sviluppo delle community del portale MyAlcoa

L'elaborazione del progetto di questa tesi, che si andrà a presentare nei prossimi capitoli, è stata sviluppata all'interno del WRC team che ha sede in Marcon -Venezia-.

## Capitolo 2

### Definizione ed analisi del progetto

---

#### 2.1 L'analisi del problema

Qualsiasi contesto aziendale utilizza parte delle proprie risorse monetarie per l'acquisto di beni utili per il raggiungimento degli obiettivi prefissati. Con il passare del tempo e con il loro utilizzo questi beni possono veder decrescere il loro valore iniziale. Nonostante ciò, continuano a rappresentare una risorsa importante per l'azienda e quindi risulta fondamentale tener traccia di tutte le loro caratteristiche.

Anche Alcoa non può fare a meno di tenere sotto controllo tali problematiche, anzi, proprio a causa delle notevoli dimensioni dell'azienda, questo processo di gestione di dati assume un'importanza non trascurabile in quanto coinvolge diversi paesi europei, e quindi persone di diversa nazionalità, oltre che a un gran numero di beni. Per poter seguire la filosofia di Alcoa di centralizzazione e standardizzazione delle informazioni, risulta quindi indispensabile un sistema informativo in grado di rispondere a tali esigenze.

### 2.1.1 La situazione attuale

Il problema di automatizzare il processo di gestione dei *Fixed Asset* nasce da una richiesta del responsabile cespiti italiano, il cui lavoro risulta spesso molto dispendioso in termini di tempo, soprattutto durante il periodo di chiusura dei conti mensili. Attualmente egli riceve mensilmente una serie di file *Excel* dal centro direzionale situato in Ungheria, ovvero un file per ogni *location* italiana. Ognuno di questi fogli elettronici contiene i dati relativi a tutti cespiti inventariati nei vari stabilimenti, quindi, come si può ben intuire, la quantità d'informazioni ricevute è notevole. Inoltre ogni riga si compone di cinquantadue colonne che rappresentano una serie di caratteristiche proprie di un cespite come il numero identificativo, la descrizione, la quantità e così via.

Ricevuti questi file, si ha la necessità di aggiungere delle informazioni addizionali rappresentate da sei colonne che descrivono alcune proprietà utili ad identificare univocamente il cespite, come il map number che descrive la posizione del cespite all'interno dello stabilimento o l'ultima data di inventario per verificare l'effettiva presenza di tale bene.

Un'altra operazione che si deve eseguire è creare un unico file che rappresenti la totalità dei cespiti italiani per poter avere una visione a livello nazionale utile per le operazioni di bilancio aziendale.

Ogni stabilimento, inoltre, ha la necessità di gestire autonomamente l'inventario dei cespiti. Questa necessità nasce dal fatto che uno stabilimento può voler tener conto di informazioni particolari diverse da quelle di altri.

Oggigiorno non esiste un metodo veloce e sicuro che permetta di gestire in maniera ottimale tale situazione, quindi si dovrà allestire un processo in grado di rispondere a queste esigenze che si possa interfacciare con il sistema informativo che sarà utilizzato.

La situazione attuale, quindi, mette in evidenza alcuni aspetti che possono essere considerati svantaggiosi:

- una gran quantità di lavoro manuale con conseguente dispendio di tempo
- molteplicità di file da aggregare con possibilità di perdita d'informazione
- assenza di *data recovery*
- difficoltà nel creare una reportistica accettabile

Da questi presupposti si è partiti per cercare di trovare una soluzione informatica in grado di gestire nel miglior modo possibile tale situazione, ampliandola a livello europeo.

## **2.2 La nuova soluzione**

Per prima cosa è stato necessario capire e definire in modo chiaro come si pensava di procedere all'inventario mensile dei cespiti. La decisione, presa ai livelli aziendali più elevati, è stata quella di acquistare tutto il necessario per poter inventariarli tramite il sistema dei codici a barre. Si è proceduto quindi all'acquisto di scanner IR per la lettura di tali codici e delle librerie richieste per la loro stampa.

Si è potuto allora procedere allo studio e alla definizione di un sistema informatico che rispondesse alle esigenze di progetto.

Alcoa possiede una Intranet, una rete privata, basata su protocolli di Internet (TCP/IP), accessibile solamente ai dipendenti che risulta uno strumento ideale per la gestione delle comunicazioni interne e come *repository* di informazioni e di documentazioni aziendali.

Le caratteristiche fondamentali di questo strumento riguardano la sicurezza, è possibile determinare bidirezionalmente il controllo degli accessi, la velocità, in quanto utilizza linee dedicate ed è quindi possibile controllare l'intensità del traffico nella rete, e la struttura gerarchica dell'informazione dato che l'accesso ai servizi di Intranet può essere reso selettivo anche rispetto alla posizione aziendale dell'utente. La Intranet quindi è il

mezzo ideale su cui “far girare” il software per la gestione dei cespiti ( vedi Figura 2.1).

L’idea di base è stata quella di realizzare un’applicazione web che sia in grado di automatizzare il più possibile le operazioni descritte nel paragrafo precedente.

Il sistema informatico chiamato *Asset Application* è stato suddiviso in tre parti principali alle quali si accede da una schermata comune che permette l’autenticazione al sistema da parte dell’utente (*Users logs in*). Le funzionalità specifiche di ognuna di esse sono:

1. Parte: “Review Data”

- Selezione della LBC (*Location Business Code*) tra le quali l’utente è abilitato
- Ricerca del cespite in base all’identificativo (TAG\_NUMBER) e alla descrizione
- Aggiornamento del cespite inserendo le informazioni addizionali

2. Parte: “Upload Data”

- Aggiornamento cespiti inventariati memorizzando la data di inventario, le note e la quantità

Queste informazioni sono rilevate da un software creato appositamente come interfaccia tra lo scanner che legge i codici a barre dei cespiti e il PC dell’utente. In pratica, si tratta di una serie di fogli excel che utilizzano delle macro per generare uno sheet formato da tre colonne (TAG\_NUMBER, NOTE, QUANTITY) nel quale sono riportati i cespiti inventariati. Da questo sheet viene effettuato un copy&paste nella *textarea* della pagina “Upload DATA” dell’applicazione che automaticamente andrà ad aggiornare il database inserendo anche la data di inventario.

3. Parte “View Data”

- Visualizzazione dati relativi alle cinquantadue colonne più le sei colonne delle informazioni addizionali

I dati possono essere visualizzati selezionando il periodo (MONTH, YEAR), lo stabilimento (LBC), il codice del libro cespiti (BOOK\_TYPE\_CODE) o, specificatamente, l’identificativo del cespite desiderato (TAG\_NUMBER).

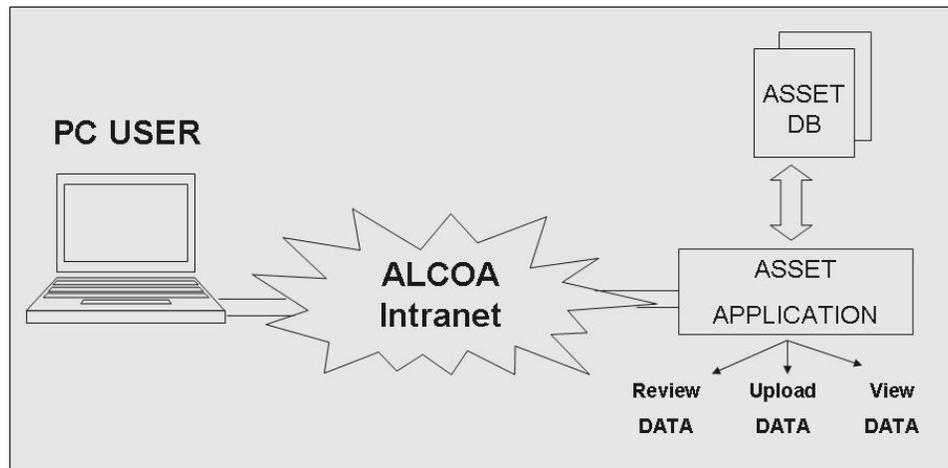


Figura 2.1 La nuova soluzione

### 2.2.1 I codici a barre

Come detto precedentemente, per l'inventario fisico dei cespiti si è reso necessario un sistema di codifica delle informazioni adatto ai sistemi automatici di rilevamento. Il metodo di utilizzare il codice a barre si è imposto nel tempo come la tecnologia vincente. Tale codifica si basa su un concetto binario, quindi è già di per se vicina al linguaggio dei calcolatori. La diversa logica di codifica, dettata da diverse esigenze applicative, ha portato a diversi tipi di codici a barre. Alcuni di essi sono ampiamente diffusi, altri vengono usati solo in speciali settori, altri ancora solo in determinati paesi. I vantaggi di utilizzare tale tipo di codifica sono:

- attendibilità: esistono due parametri, la cifra di controllo (check digit) ed un numero di digits fisso, che fanno ottenere una altissima attendibilità delle informazioni acquisite
- automazione: mediante l'utilizzo di lettori, decodificatori e scanner industriali vi è la possibilità di leggere il codice a barre in modo automatico

- precisione: i sistemi basati sul codice a barre lavorano senza errori
- velocità: l'assunzione dei dati da parte del calcolatore avviene senz'altro molto più velocemente che in modo manuale

La scelta di Alcoa è stata quella di utilizzare un codice che sia il più standard possibile a livello europeo. A tal proposito esistono dei codici regolamentati dal consorzio EAN (*European Article Number*) che rispondono perfettamente all'esigenza di uniformità ricercata. La decisione, quindi, è stata quella di avvalersi dell' EAN-128, certamente il più moderno e "factotum" tra i codici a barre di libero uso, infatti permette un'elevata densità di dati, la codifica di caratteri alfanumerici ed una buona affidabilità.

La codifica dell'EAN-128 è definita a livello internazionale allo scopo di dare uno standard comune per la codifica dei più svariati tipi di dati, rappresentando, inoltre, ciò che questi dati simboleggiano.



**Figura 2.2 Esempio di un codice a barre in Alcoa**

### 2.2.2 Finalità e Vantaggi

L'applicazione web *Asset Application* rappresenta la nuova soluzione che deve rispondere alle esigenze di progetto fin qui evidenziate. La struttura descritta nei paragrafi precedenti è stata elaborata cercando di ottimizzare i tempi di lavoro da parte dell'utente, semplificando le loro mansioni e rendendo il più possibile automatizzato il processo di gestione dei cespiti.

I vantaggi che questo prodotto genererà possono essere così evidenziati:

- *single repository* di informazioni
- *data recovery* più immediato da parte degli utenti finali
- *reporting* più semplice e più sicuro
- scalabilità del processo, in quanto l'architettura sarà in grado di far fronte ad aumenti di carico del sistema
- diminuzione del lavoro manuale da parte dell'utente con la conseguente riduzione del rischio di errore umano
- miglioramento delle *performances* del processo in termini di velocità e sicurezza
- mantenimento di uno storico per l'inventario

La realizzazione dell'applicazione è andata di pari passo con l'accertamento della correttezza delle sue funzionalità attraverso il confronto con il responsabile del progetto. Ogni componente dell'applicazione è stata definita a priori e, una volta realizzata, è stata verificata. Tale accertamento è necessario nella realizzazione di un qualsiasi progetto per confrontare il lavoro svolto con le esigenze richieste.

Una volta definite le idee di partenza e studiato in tutti i dettagli il progetto, si è potuto procedere all'elaborazione vera e propria dell'applicazione, non prima però di aver definito le scelte tecnologiche che l'azienda adotta per questo tipo di sviluppo software.

I prossimi paragrafi andranno ad illustrare queste scelte.

## 2.3 Le scelte tecnologiche

La diffusione di Internet come mezzo di comunicazione ha favorito lo sviluppo di uno strumento in grado di erogare servizi attraverso un nuovo tipo di applicazioni distribuite: le applicazioni *enterprise*. Queste implicano l'accesso ad altre applicazioni, dati o servizi sparsi all'interno dell'infrastrutture informatiche di un'azienda e quindi rappresentano una fonte essenziale a cui accedere per ogni tipo di operazione da eseguire su tali dati.

Inoltre consentono una maggior facilità di distribuzione dovuta al fatto che un'applicazione pubblicata sul server è subito fruibile da tutti gli utenti, un accesso multiplatforma indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti, e una riduzione dei costi di gestione.

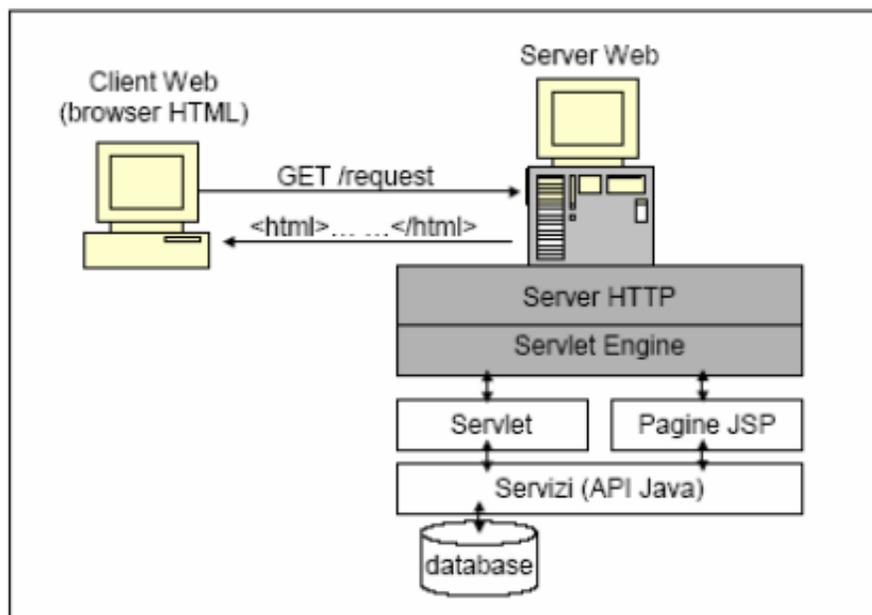
### 2.3.1 Architettura di un'applicazione web

Un'applicazione Web si basa su elementi software standard indipendenti dalle caratteristiche della particolare applicazione e dalla piattaforma software e hardware su cui viene eseguita. Nella maggior parte dei casi si sviluppa su tre livelli logico-funzionali:

- livello di presentazione: funzionalità legate all'interazione con l'utente. E' costituito da vari componenti combinati tra di loro (browser, documenti HTML, Applet Java, controlli ActiveX)
- livello intermedio: racchiude la logica elaborativa dell'applicazione e spesso dipende dalle caratteristiche e dalle tecnologie supportate dal server Web. Può essere costituito da un insieme di script e programmi interagenti tra di loro tramite specifiche tecnologie
- livello dati: fornisce servizi non direttamente disponibili dal server Web. Tipici esempi di applicazioni presenti a questo livello sono i server dati (DBMS) e i server di mail.

Il linguaggio scelto per la realizzazione dell'applicazione è Java perché rappresenta la scelta ideale per un'architettura server-side, in quanto la sua caratteristica di portabilità su ogni tipo di piattaforma di sviluppo permette l'utilizzo di un'applicazione su ogni tipo di ambiente server-side esistente.

Per dare la possibilità a tutti di poter usufruire delle potenzialità di Java, la Sun ha prodotto una completa piattaforma di sviluppo chiamata *Java 2 Platform Enterprise Edition (J2EE)*, un insieme di specifiche per la realizzazione di un software enterprise. L'intera applicazione è stata sviluppata utilizzando la tecnologia dell'architettura J2EE: *Java Server Pages*.



**Figura 2.3** Architettura di un'applicazione web basata su Servlet, Jsp, Java API

### 2.3.2 Servlet, JSP e Tomcat

La crescente richiesta di pagine web con contenuti dinamici e di servizi su web, ha portato allo sviluppo di una particolare tipologia di software che collabora con il server web per estenderne le funzionalità e poter interagire con basi di dati.

Le Servlet sono moduli software scritti in Java che vengono eseguiti in applicazioni lato server per esaudire le richieste dei client. Per scrivere una Servlet si fa uso delle classi del package `javax.servlet` che è il framework di base per la scrittura delle servlet e del package `javax.servlet.http` che è l'estensione del framework di base, per la comunicazione via http. Utilizzando un linguaggio portabile come Java, le Servlet consentono di realizzare soluzioni per estendere le funzionalità dei server web, indipendenti dal sistema operativo su cui esse vengono eseguite.

Le servlet vengono caricate solo una volta, al momento della prima richiesta, e rimangono residenti in memoria pronte per servire le richieste fino a quando non vengono chiuse, con ovvi vantaggi in fatto di prestazioni. Inoltre sono facili da utilizzare, portabili ed economiche in quanto molti server web che supportano le servlet sono free o comunque hanno un basso costo.

Sullo stesso livello delle servlet (come da Figura 2.3) viene considerata la tecnologia **JSP (*Java Server Pages*)**, una soluzione multiplatforma per realizzare pagine HTML dinamiche dal lato server. Questa tecnologia agli occhi del programmatore viene gestita per mezzo di un linguaggio di script che è in grado di mescolare codice HTML, componenti riusabili (JavaBeans), applicazioni remote (Servlet) e codice Java. Normalmente una Pagina JSP è composta da:

- porzioni di codice HTML, JavaScript, CSS e così via, non compilati dal motore Jsp (Jsp Engine) e comunemente definiti *blocchi di codice statico*
- porzioni di codice Java, compilati dal motore Jsp, che prendono il nome di *blocchi di codice dinamico*.

La prima volta che un Client effettua la richiesta di una Pagina JSP ad un Web Server (*container JSP*), questo la compila creando un oggetto Servlet che risiederà in memoria. Solo dopo questo passaggio l'output viene inviato al Client che potrà interpretarlo come se fosse una semplice Pagina HTML. Ad ogni richiesta successiva della medesima Pagina JSP, il Web Server controlla se è stata modificata: in caso negativo richiama il Servlet già compilato, altrimenti si occupa di eseguire nuovamente la compilazione e memorizzazione del nuovo Servlet. Conseguenza di questo meccanismo è che la prima volta che si invoca una Pagina JSP il sistema sarà un pò più lento del normale, mentre dalle invocazioni successive si avranno prestazioni più veloci.

Le pagine JSP non possono venire visualizzate mediante l'utilizzo di un normale server web, necessitano infatti di un particolare software in grado di interpretarle. L'implementazione di riferimento per le specifiche delle servlet Java e delle JSP è **Tomcat**, il servlet container open source sviluppato dal progetto Apache Jakarta.

Tomcat contiene al suo interno tutte le funzionalità tipiche di un web server, ovvero ha la capacità di interpretare una richiesta di una risorsa veicolata su protocollo HTTP, indirizzarla ad un opportuno gestore (o prenderla dal filesystem) e restituire poi il risultato (codice HTML o contenuto multimediale che sia). In questo aspetto non differisce molto da altri web server, come Apache, se non che spesso si rivela meno prestante. La caratteristica di Tomcat non è quindi quella di essere un veloce web server, ma piuttosto quella di fornire allo sviluppatore un vero e proprio ambiente nel quale girano applicazioni Java.

Come nelle normali applicazioni, anche nella programmazione lato server gli oggetti creati devono avere uno *scope*, ovvero essere definiti ed accessibili in un determinato contesto. Utilizzare Tomcat equivale a suddividere lo spazio delle risorse in **contenitori** sostanzialmente indipendenti nei quali possono essere depositati oggetti Java, secondo una logica orientata al web.

Per comprendere appieno questo concetto analizziamo nel dettaglio i singoli contenitori che sono tra loro strutturati in maniera abbastanza gerarchica.

- Tomcat: il primo livello corrisponde al web server stesso che avrà un insieme di variabili ad esso associate. Ad esempio qui dentro vi sarà indicazione del numero di thread attualmente in esecuzione, della versione di Tomcat o altre proprietà globali generalmente di poco interesse nella programmazione
- Contesto (*context*): all'interno di questo contenitore si troveranno quindi oggetti comuni a tutta la sezione, che saranno gli stessi per tutti gli utenti e tutte le applicazioni del contesto. Questo contenitore viene generato all'avvio di Tomcat e rimane attivo indipendentemente dal fatto che qualcuno lo stia interrogando in quel momento. Questo non vuol dire che rimanga invariato, anzi, il suo contenuto cambierà in funzione delle richieste ricevute
- Sessione (*session*): questo contenitore viene associato ad un utente (o, meglio, ad un client) per tutto il tempo per il quale rimane connesso ad un contesto specifico. Al suo interno verranno memorizzati oggetti che saranno accessibili solo a quell'utente e ne personalizzeranno l'ambiente rispetto ad altri client che utilizzino risorse dello stesso contesto nello stesso momento
- Richiesta (*request*): è il contenitore con la vita più breve che nasce e muore nell'ambito della singola transazione web. Raramente questo contenitore viene riempito da applicazioni anche se svolge un ruolo fondamentale nel flusso dei dati che attraversano Tomcat.

Se un client accede a una risorsa su di un server su cui è attivo Tomcat, innanzi tutto viene istanziato un oggetto *request* nel quale vengono inserite tutte le informazioni ricevute tramite HTTP e "al suo fianco" viene istanziato un oggetto risposta (*response*) nel quale costruire la risposta da restituire al client.

### 2.3.3 RDBMS Oracle

Un DBMS (*Data Base Management System*) è un sistema di gestione di basi di dati costituito, in genere, da più programmi informatici che controllano l'organizzazione, la memorizzazione e il reperimento dei dati (campi, record e archivi) in un database.

Esso controlla anche la sicurezza e l'integrità del database e accetta richieste di dati da parte del programma applicativo istruendo il sistema operativo per il trasferimento dei dati appropriati.

La versione di Oracle attualmente utilizzata in Alcoa è *Oracle Server Release 9i Enterprise Edition* che si appoggia a database di tipo relazionale. Questo particolare sistema di gestione è anche chiamato RDBMS (*Relational Data Base Management System*). Quando si usa un RDBMS i sistemi informativi possono essere adeguati molto facilmente al cambiamento delle richieste informative dell'organizzazione. Possono essere aggiunte al database nuove categorie di dati senza dover stravolgere il sistema esistente.

Il sistema di sicurezza dei dati impedisce agli utenti non autorizzati di visualizzare o aggiornare il database. Mediante l'uso di *password* agli utenti è permesso l'accesso all'intero database o ad un suo sottoinsieme.

Un RDBMS fornisce un modo per aggiornare ed immettere nuovi dati nel database, oltre che per interrogarlo. I linguaggi di interrogazione del database mediante *queries* e i generatori di *report* permettono agli utenti di interrogare in maniera interattiva il database e di analizzarne i dati. Il linguaggio di interrogazione normalmente utilizzato è SQL.

L'RDBMS Oracle offre particolari funzionalità rispetto ad altri RDBMS, come la caratteristica di determinare il percorso più efficiente per accedere ai dati, la possibilità di scrivere in un unico blocco diversi statement SQL e di passarli ad Oracle tutti assieme (PL/SQL) o l'opzione di poter creare dei *trigger*, procedure registrate nel database e automaticamente eseguite quando questo viene modificato.



## Capitolo 3

### La realizzazione del progetto

---

#### 3.1 Definizione dei requisiti funzionali

Una volta chiarite le esigenze del committente attraverso dei *meeting* organizzati, o tramite *conference call*, e specificate le operazioni che potranno essere eseguite con la nuova soluzione, si è potuto procedere all'analisi dettagliata dei requisiti funzionali che l'applicazione dovrà avere per rispondere a tali richieste. .

##### 3.1.1 Autenticazione utenti e privilegi

La prima necessità è stata quella di distinguere l'accesso all'applicazione tra due tipologie di utente. L'autenticazione (*Login*) avviene tramite l'inserimento di *username* e *password* Cii (*Common Information Infrastructure*), uno standard aziendale che Alcoa ha stabilito nel 1994 a livello di hardware e di software per la condivisione a livello globale di informazioni.

Questa operazione è la stessa per tutti gli utenti e, in generale, viene utilizzata anche per tutte le applicazioni web sviluppate dal WRC.

Dopo l'autenticazione è necessario un controllo sul tipo di utente che sta effettuando l'accesso al sistema. La richiesta era di poter permettere ad alcuni cosiddetti *superuser* di visualizzare tutte le LBC relative al paese di appartenenza e quindi di poter gestire le informazioni sui cespiti a livello nazionale. Gli altri utenti, invece, dovevano avere la possibilità di accedere solamente alle informazioni relative alla loro LBC.

La soluzione pensata e realizzata per adempiere a questa esigenza è stata quella di creare un attributo "LBC" da associare ad ogni utente. In questo campo viene specificato il codice della location di appartenenza nel caso di normal user, mentre, nel caso di superuser, viene inserita una particolare decodifica così strutturata:

- ALL-XX dove XX rappresenta il codice del paese (ad esempio ALL-IT,...)

In questo modo, con una particolare query che sarà descritta nei prossimi paragrafi, è possibile estrarre tutte le LBC relative a quel paese.

Un altro requisito funzionale a livello di utente è rappresentato dalla necessità di eseguire correttamente l'uscita dall'applicazione, quindi si è provveduto a realizzare un opportuno bottone di *logout* che permette l'esecuzione di questa operazione nel giusto modo, chiudendo esplicitamente la propria sessione di lavoro.

Strettamente collegato con le funzionalità del logout, è stato creato un processo di controllo della sessione, inserito in ogni pagina dell'applicazione, che verifica lo username dell'utente. Se, ad esempio, l'applicazione è aperta e non viene utilizzata per un certo tempo (*time-out*) la sessione scade (*expired*), obbligando l'utente a rieseguire l'operazione di login.

### 3.1.2 Operazioni eseguibili

Come detto nel capitolo precedente l'applicazione è suddivisa in tre parti, ognuna delle quali deve rispondere a determinati requisiti. Cerchiamo ora di analizzarle singolarmente:

- Review DATA: questa sezione dell'applicazione riceve dalla fase di autenticazione le informazioni relative all'utente che ha eseguito l'accesso, e quindi, come detto nel paragrafo precedente, saranno già disponibili le LBC a cui esso può accedere. L'utente potrà immettere dei parametri per ricerca dei cespiti e visualizzare tali informazioni.

In questo contesto è stata richiesta una particolarità per la gestione italiana.

In Italia esistono due LBC di *look-up* (una relativa ad Alcoa Trasformazioni e una ad Alcoa Servizi), cioè che non corrispondono effettivamente a degli stabilimenti esistenti, ma contengono dei cespiti che possono appartenere a diverse LBC. Si tratta di beni acquistati prima del 1996 e che è possibile ricondurre alla effettiva LBC di appartenenza in base al contenuto di una specifica colonna della tabella XXFA\_ASSET\_LISTING (ATTRIBUTE13). I cespiti nei quali questo campo risulta vuoto e il campo relativo al codice del libro ha un particolare valore si riferiscono a un'LBC specifica (una per ATR e una per ASV).

Questa esigenza ha sicuramente reso più complessa la query di estrazione dei dati.

Una volta selezionata la LBC desiderata viene visualizzata la tabella dei cespiti esistenti mettendo in evidenza il TAG\_NUMBER, la descrizione e la quantità. Ognuno di essi può essere aggiornato, inserendo o modificando solamente le informazioni addizionali relative.

- Upload DATA: la parte di upload dei dati relativi ai cespiti inventariati si interfaccia, come detto, ad un sistema software costituito da alcuni fogli excel realizzato appositamente per tale scopo. Le funzionalità che questa parte dell'applicazione deve possedere si limitano al controllo della correttezza delle

informazioni inserite, e quindi, all'aggiornamento del database con l'aggiunta della data di inventario.

- View DATA: l'ultima parte fa riferimento alla query SQL realizzata attraverso un particolare software (ART) che permette la visualizzazione dei dati via web, oppure l'*export* degli stessi in formato excel.

I requisiti funzionali che questa query deve avere sono la possibilità di selezione dei parametri di ricerca (COUNTRY, LBC, BOOK\_TYPE\_CODE, MONTH, YEAR) e la visualizzazione delle cinquantadue colonne provenienti dal database di EBS con l'aggiunta delle sei colonne aggiuntive tramite un *join* tra le due tabelle.

### 3.2 Il Database

L'applicazione utilizza dei dati che sono già presenti in una tabella creata appositamente per la gestione dei fixed asset chiamata XXFA\_ASSET\_LISTING. Essa è formata da circa settanta colonne tra le quali anche le cinquantadue che saranno utilizzate nella visualizzazione che il committente ha richiesto. In più ne utilizza un'altra per estrarre le LBC del paese di appartenenza degli superuser (XXEBS\_LBC\_FULL\_HIERARCHY).

La necessità è stata quella di creare una tabella aggiuntiva nella quale poter inserire i dati relativi alle informazioni addizionali, oltre, ovviamente, alla tabella che gestisce l'accesso all'applicazione identificando il tipo di utente che sta eseguendo l'autenticazione.

Inoltre, in seguito alla richiesta di tener memorizzato uno storico relativo alle ultime due date di inventario di un cespite, si è resa necessaria la creazione di un'ulteriore tabella contenente l'identificativo del cespite e, appunto, le relative ultime due date di inventario.

Per eseguire automaticamente quest'ultimo punto si è pensato di utilizzare un *trigger*, una sorta di procedura particolare che si attiva dopo un determinato evento. Gli eventi per i quali si attiva un trigger sono l'esecuzione di una istruzione INSERT / UPDATE / DELETE su di una tabella. Nel nostro caso, risulta uno strumento molto utile da utilizzare ogniqualvolta viene eseguito un UPDATE sull'ultima data di inventario, in modo da mantenere il dato precedente per la costruzione dello storico.

### 3.2.1 La progettazione del database

La progettazione di un database è generalmente articolata in tre fasi successive:

- progettazione concettuale: rappresentazione delle specifiche informali sotto forma di schema concettuale; descrizione formale e completa, che fa riferimento ad un modello concettuale il cui obiettivo è la rappresentazione del contenuto informativo della base di dati
- progettazione logica: traduzione dello schema concettuale nello schema logico; fa riferimento al modello logico dei dati prescelto, si usano criteri di ottimizzazione delle operazioni da fare sui dati, la qualità dello schema va verificata mediante tecniche formali (normalizzazione)
- progettazione fisica: specifica dei parametri fisici di memorizzazione dei dati (organizzazione dei file e degli indici), produce un modello fisico, che dipende dal DBMS prescelto



**Figura 3.1 La progettazione di un database**

**PROGETTAZIONE CONCETTUALE**

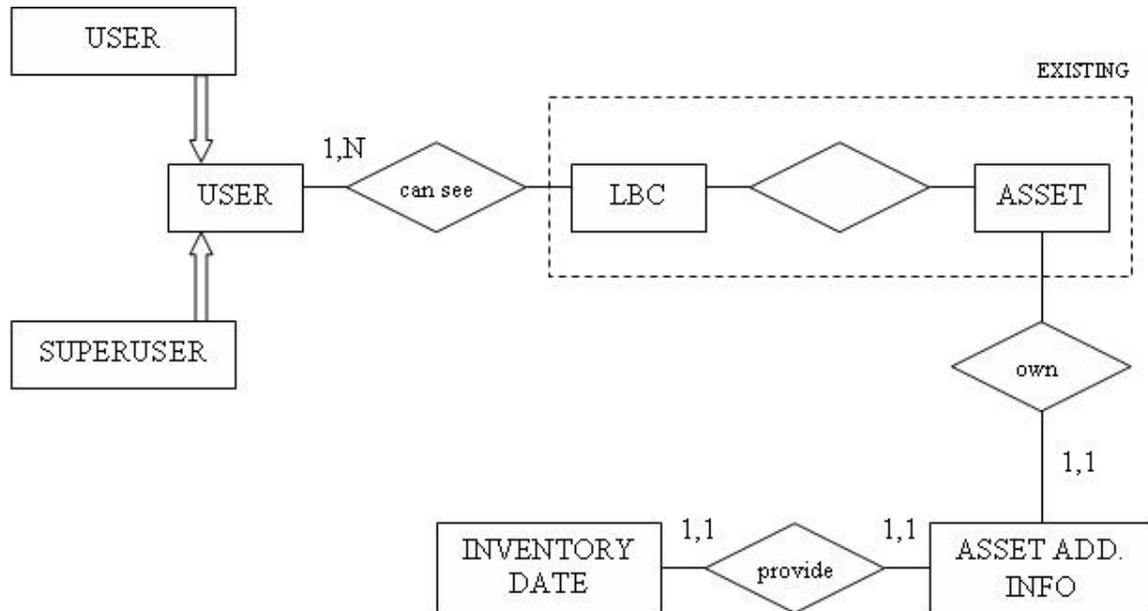


Figura 3.2 Schema concettuale

**Entità**

- User: descrive le caratteristiche principali che devono essere registrate affinché l'utente possa autenticarsi al sistema. E' formata da due sottoclassi.
- Asset Additional Info: descrive le informazioni addizionali del cespite e memorizza l'utente che le ha inserite. La chiave è TAG\_NUMBER.
- Inventory Date: memorizza le ultime due date di inventario di un cespite. E' aggiornata mediante l'utilizzo di un trigger e la chiave è TAG\_NUMBER.

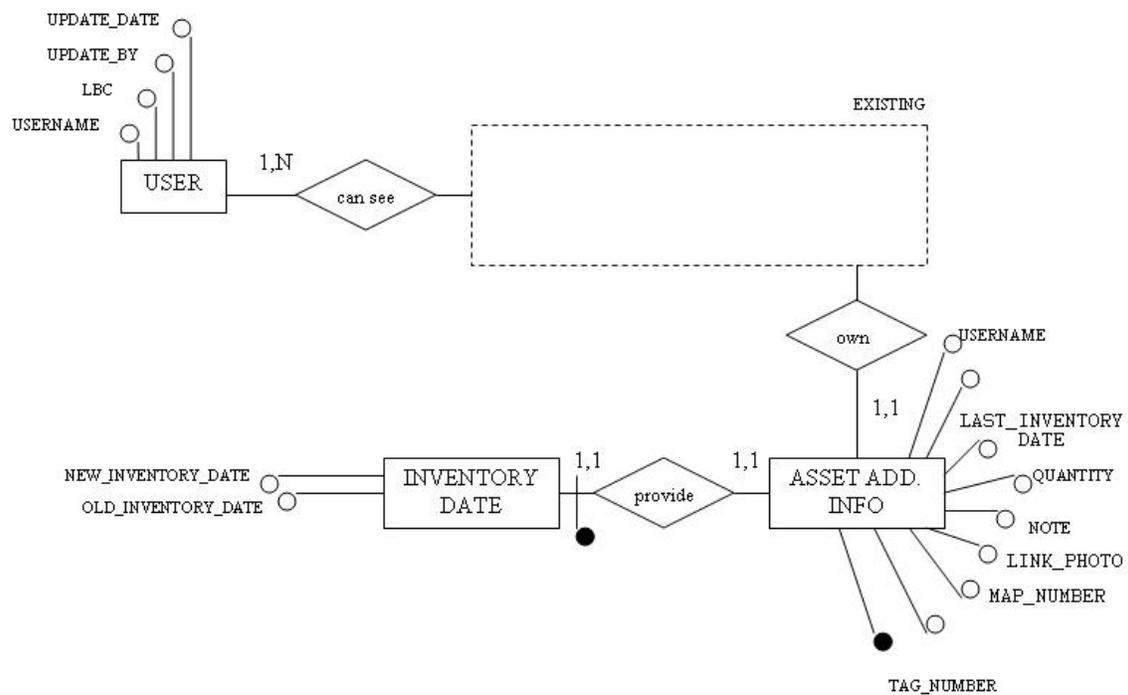
**Associazioni**

- Can see: permette l'accesso solo alle LBC consentite
- Own: assegna al cespite le additional info che esso possiede
- Provide: fornisce l'ultima data di inventario del cespite

**PROGETTAZIONE LOGICA**

La traduzione verso il modello logico passa attraverso la fase di ristrutturazione dello schema concettuale che può essere suddivisa in una serie di passi da effettuare in sequenza. Nel nostro caso il passo più importante consiste nell'eliminazione della generalizzazione della classe USER. La soluzione adottata è stata quella di accorpare le entità figlie nell'entità padre, aggiungendo a quest'ultima un particolare attributo (LBC) in grado di identificare le due sottoclassi. Questa ristrutturazione potrebbe causare delle ridondanze in quanto potrebbero esserci più tuple relative allo stesso username (non nel caso di superuser). Ma, dopo un consulto con il committente, è stato ritenuto il modo migliore di proseguire, in quanto un utente può avere al massimo due LBC a cui è associato (una per Alcoa Trasformazioni e una per Alcoa Servizi).

L'ultimo passo per la definizione della schema logico è quello di definire gli identificatori primari per le entità. Lo schema risultante è riportato in Figura 3.3.



**Figura 3.3 Schema logico**

## **PROGETTAZIONE FISICA**

```
CREATE TABLE XXFA_WEBASSET_USER_LBC (  
    USERNAME          VARCHAR(20),  
    LBC               VARCHAR(6), -- contain the LBC value or the keyword  
                                ALL-XX, where XX is the country code  
    UPDATE_BY        VARCHAR(20),  
    UPDATE_DATE      DATE  
);
```

```
CREATE TABLE XXFA_WEBASSET_INFO(  
    TAG_NUMBER        VARCHAR(20) NOT NULL PRIMARY KEY,  
    DEPART_RESP_AREA VARCHAR(30),  
    MAP_NUMBER        VARCHAR(30),  
    LINK_PHOTO        VARCHAR(50),  
    NOTE              VARCHAR(150),  
    QUANTITY          VARCHAR(5),  
    LAST_INVENTORY_DATE DATE,  
    CREATION_DATE     DATE,  
    USERNAME          VARCHAR(20)  
);
```

```
CREATE TABLE XXFA_WEBASSET_UPDATE_HISTORY(  
    TAG_NUMBER        VARCHAR(20) NOT NULL PRIMARY KEY,  
    OLD_INVENTORY_DATE DATE,  
    NEW_INVENTORY_DATE DATE  
);
```

```
CREATE TRIGGER XXFA_WEBASSET_TRIGGER_DATE  
BEFORE UPDATE OF LAST_INVENTORY_DATE ON XXFA_WEBASSET_INFO  
REFERENCING NEW AS n OLD AS o  
FOR EACH ROW  
    BEGIN  
        UPDATE XXFA_WEBASSET_UPDATE_HISTORY SET  
            OLD_INVENTORY_DATE=:o.LAST_INVENTORY_DATE,  
            NEW_INVENTORY_DATE=:n.LAST_INVENTORY_DATE  
        WHERE TAG_NUMBER=:o.TAG_NUMBER;  
  
        IF( SQL%ROWCOUNT = 0) THEN  
            INSERT INTO XXFA_WEBASSET_UPDATE_HISTORY  
                (TAG_NUMBER, OLD_INVENTORY_DATE, NEW_INVENTORY_DATE)  
            VALUES (  
                :o.TAG_NUMBER,  
                :o.LAST_INVENTORY_DATE,  
                :n.LAST_INVENTORY_DATE);  
        END IF;  
    END;
```

### **3.3 L'implementazione e il layout**

In questo paragrafo andremo ad analizzare l'applicazione nel dettaglio. In particolare verrà visualizzato il *layout* delle pagine, ossia il prototipo realizzato come interfaccia per l'utente, ed alcune porzioni di codice particolari necessari per la corretta implementazione delle funzionalità che l'applicazione deve avere.

Il prototipo è realizzato con pagine HTML statiche e fogli di stile (CSS) per separare i contenuti dalla presentazione e rendere più leggibile il codice per eventuali modifiche da apportare.

L'implementazione delle funzionalità, invece, sono realizzate da pagine JSP nelle quali possono trovarsi sia parti di codice HTML statico che variabili contenenti linguaggio SQL per l'interrogazione o l'aggiornamento del database.

### 3.3.1 Autenticazione

L'accesso all'applicazione è consentito solamente ad utenti che sono registrati nella tabella XXFA\_WEBASSET\_USER\_LBC. L'operazione di accesso viene eseguita digitando nell'apposito *form* il proprio username e la password. Il campo *domain* rappresenta il dominio intranet di appartenenza e di default è EUROPE, in quanto l'applicativo è utilizzato da utenti europei.

Le informazioni inserite dall'utente vengono verificate da una pagina che non fornisce output all'utente che è stata chiamata *execValidateNTLogin.jsp*. La validazione avviene mediante la classe *validateNTLogin* che è stata creata appositamente dal Web Team per essere utilizzata in tutte le applicazioni web che necessitano di un'autenticazione. Questa classe verifica se si tratta di un utente Cii valido, ma per accedere è necessario anche essere registrati nella tabella degli utenti dell'applicazione. Questo controllo è stato eseguito con la seguente query:

```
SELECT USERNAME FROM XXFA_WEBASSET_USER_LBC WHERE USERNAME=?
```

Se questi controlli vanno a buon fine l'utente può accedere all'applicativo, altrimenti si ritorna alla pagina di login che visualizzerà il tipo di errore verificatosi.

La pagina iniziale è la seguente:

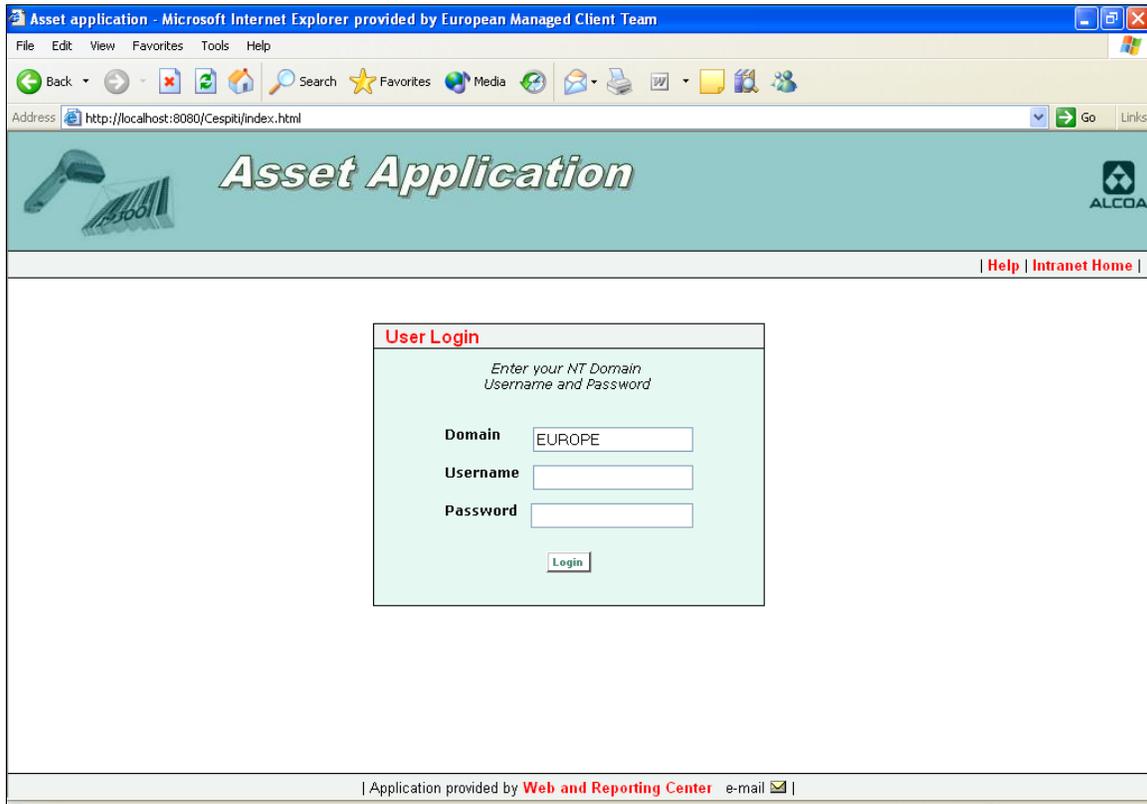


Figura 3.3 La pagina iniziale: User Login

I messaggi di errore che si possono verificare sono:

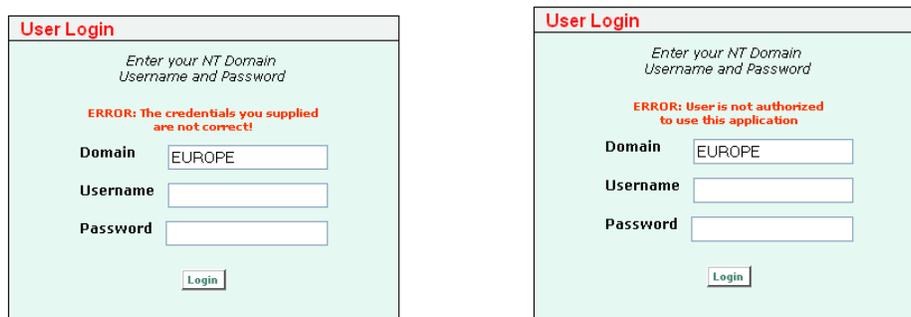


Figura 3.4 La pagina iniziale: Messaggi di errore

Capitolo 3. La realizzazione del progetto

---

Dalla fase di login vengono inoltre eseguite altre due importanti procedure:

- l'inizializzazione del pool di connessioni al database
- la scrittura nel file di log per la memorizzazione degli accessi all'applicazione

La prima avviene mediante l'utilizzo del metodo *init()* richiamato dalla servlet generata dal web container al momento della richiesta. Il pool di connessioni verrà poi chiuso con il metodo *destroy()*. Questi due metodi sono stati definiti nel file di inizializzazione *dataBaseInit.jsp* come riportato di seguito.

```
public void jspInit()
{
    ServletContext context = getServletContext().getServletContext();
    String jdbc_url = getServletContext().getInitParameter("jdbc_url");
    String Asset_username =
    getServletContext().getInitParameter("Asset_username");
    String Asset_password =
    getServletContext().getInitParameter("Asset_password");

    db = (dataBase) context.getAttribute("Asset_db");

    if (db == null ) {
        String DRIVER = "oracle.jdbc.driver.OracleDriver";
        try {
            Class.forName(DRIVER);
        } catch (ClassNotFoundException ex) {
            System.err.println("JDBC Driver Class Not Found");
        } catch (Exception ex) {
            System.err.println("Exception" + ex);
        }
    }
    db = new dataBase();
    db.setName("Asset_db");
    db.setUrl(jdbc_url);
    db.setUsername(Asset_username);
    db.setPassword(Asset_password);
    db.setLogFileName("Asset.log");
    context.setAttribute("Asset_db", db);
}

public void jspDestroy()
{
    db.close(); // close all open connections at shutdown
}
```

Per tutte le altre pagine dell'applicazione si richiederà la connessione al Pool e non direttamente al database Oracle attraverso il seguente comando che rileva la connessione dal contesto:

```
ServletContext context = getServletConfig().getServletContext();
db = (dataBase) context.getAttribute("Asset_db");
Connection con = db.getConnection();
```

La scrittura nel file di log è legata a motivi di sicurezza interna e viene eseguita ogni volta che si tenta di fare un accesso all'applicazione. Essa avviene tramite una funzione richiamata nel file *execValidateNTLogin.jsp* come segue:

```
String stringLg = new String(new java.util.Date().toString()+"#L_Y#" +
username + " - " + request.getRemoteAddr() + " # SUCCESSFULLY AUTH\n");
writeLog(stringLg);
```

La funzione che permette di scrivere nel file log è la seguente:

```
final String sep = java.io.File.separator;

public synchronized boolean writeLog(String str)
{
    ServletContext ctx = getServletConfig().getServletContext();
    String logfilename =
        getServletContext().getRealPath(".") + sep + "log" + sep + "Asset.log";

    try{
        FileOutputStream logfile = new FileOutputStream(logfilename, true);
        byte[] buf;
        buf = new byte[str.length()];
        buf = str.getBytes();
        logfile.write(buf);
        logfile.close();
    }
    catch (IOException e) {
        e.printStackTrace();
        return false;
    }
    catch (Exception ex) {
        ex.printStackTrace();
        return false;
    }
    return true;
}
```

Capitolo 3. La realizzazione del progetto

---

### 3.3.2 Review DATA

La parte di “Review DATA” è quella a cui si accede automaticamente dopo il login, oppure attraverso un click sul bottone omonimo. Essa si divide a sua volta in varie fasi: da un form di ricerca si passa alla visualizzazione della ricerca stessa e infine alla possibilità di aggiornare le informazioni aggiuntive del cespite selezionato.

Nella prima fase viene eseguita una query sulla tabella XXEBS\_LBC\_FULL\_HIERARCHY, la quale permette di estrarre tutte le LBC di un superuser oppure, tramite la tabella XXFA\_WEBASSET\_USER\_LBC, vengono visualizzate solo le LBC alle quali un normal user è abilitato. La query è la seguente:

```
SELECT U.LBC_DESCR,U.LBC FROM WEBAPP.XXEBS_LBC_FULL_HIERARCHY U
WHERE
LBC IN (SELECT LBC FROM XXFA_WEBASSET_USER_LBC WHERE USERNAME=?)
OR
U.COUNTRY IN (SELECT SUBSTR(LBC,5,6) FROM XXFA_WEBASSET_USER_LBC
WHERE USERNAME=? AND SUBSTR(LBC,1,3) =?)
```

Il form permette la selezione della LBC desiderata e la possibilità di inserire dei parametri di ricerca (TAG\_NUMBER e DESCRIPTION). Si presenta come segue:

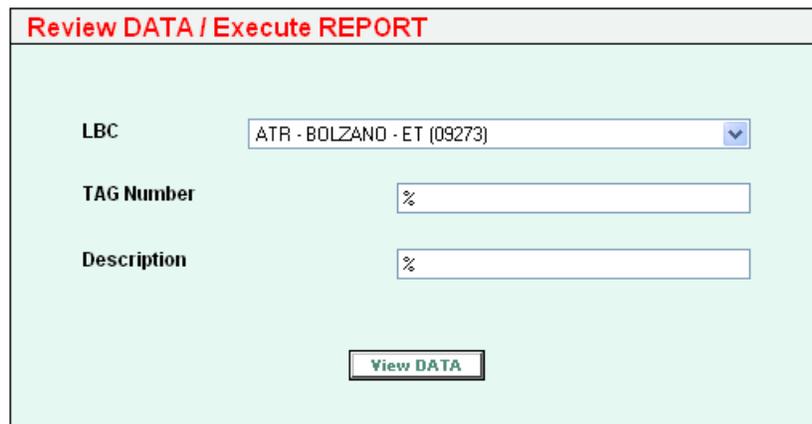


Figura 3.5 Il form di ricerca

Dopo aver inserito i parametri di ricerca e aver selezionato la LBC viene visualizzata la lista di cespiti presenti, eseguendo una query abbastanza complessa a causa dell'esigenza di integrare anche quelli delle LBC che fungono da valori di look-up per gli altri.

Il committente ci ha fornito una codifica che permette di associare il cespite alla corretta LBC. Questa è stata eseguita con il comando SQL *decode*, il quale funziona nel seguente modo:

```
DECODE(expr,search1,return1, [search1,return1,...][default])
```

se expr eguaglia qualche search, restituisce return, altrimenti restituisce default

La query, composta dall'unione di tre sottoquery, è la seguente:

```
SELECT TAG_NUMBER,DESCRIPTION,UNITS FROM XXFA.XXFA_ASSET_LISTING
WHERE TAG_NUMBER LIKE ? AND LBC=? AND DESCRIPTION LIKE UPPER(?)
AND MONTH=TO_CHAR( ADD_MONTHS (SYSDATE,-1) , 'MM')
AND YEAR=TO_CHAR( ADD_MONTHS (SYSDATE,-1) , 'YYYY')
AND BOOK_TYPE_CODE LIKE DECODE(COUNTRY, 'IT', '%_F', '%_C')

UNION ALL

SELECT TAG_NUMBER,DESCRIPTION,UNITS FROM XXFA.XXFA_ASSET_LISTING
WHERE TAG_NUMBER LIKE ?
AND
SUBSTR(LBC,2,5)||ATTRIBUTE13=DECODE(?,'01266','9295U126','09211',
'9295U921','09221','9295U9221','09222','9295U9222','09243',
'9295U924','09253','9295U925','09273','9295U927','19234',
'9295BAREGGIO','19211','9295UF72','19222','9295UF74','19221',
'9295UF74-PRIMARY','19253','9295UF76','19273','9295UF77',
'11266','9295UF79','19243','9295UF81','19205','9295UF82')
AND DESCRIPTION LIKE UPPER(?)
AND MONTH=TO_CHAR( ADD_MONTHS (SYSDATE,-1) , 'MM')
AND YEAR=TO_CHAR( ADD_MONTHS (SYSDATE,-1) , 'YYYY')
AND BOOK_TYPE_CODE LIKE '%_F'

UNION ALL

SELECT TAG_NUMBER,DESCRIPTION,UNITS FROM XXFA.XXFA_ASSET_LISTING
WHERE TAG_NUMBER LIKE ?
AND LBC=DECODE(?,'27856','09295','17856','19295')
AND DESCRIPTION LIKE UPPER(?)
AND MONTH=TO_CHAR( ADD_MONTHS (SYSDATE,-1) , 'MM')
AND YEAR=TO_CHAR( ADD_MONTHS (SYSDATE,-1) , 'YYYY')
AND BOOK_TYPE_CODE LIKE '%_X'
```

Dalla tabella dei cespiti che si va a creare è possibile, cliccando sul bottone “Update”, andare a visualizzare ed, eventualmente aggiornare, le informazioni aggiuntive relative a quel cespite.

**Report / Review results**

Tag Number	Description	Units	Update
0927912-G00500	ACQUISTO PROIETTORE E FOTOCAMERA	1	<input type="button" value="Update"/>
0927912AG00500	FOTOCAMERA OLYMPUS C860L MATR. 70105196	1	<input type="button" value="Update"/>
0927631BG02101	N. 2 CONTROPUNZIONI+CAMICIA INTERNA OCNTENITORE PRESSA 3500	1	<input type="button" value="Update"/>
0929641-J61101	CAMERE DI RAFFREDDAMENTO ORMIS/ALSTEP-UP ALLOCATION	2	<input type="button" value="Update"/>
0929641-J61100	CAMERE DI RAFFREDDAMENTO ORMIS/AL	2	<input type="button" value="Update"/>
0929641-J61001	CAMERA DI RAFFREDDAMENTO SITISTEP-UP ALLOCATION	1	<input type="button" value="Update"/>
0929641-J61000	CAMERA DI RAFFREDDAMENTO SITI	1	<input type="button" value="Update"/>

**Figura 3.6 La tabella dei cespiti**

Vengono visualizzati il TAG\_NUMBER e la DESCRIPTION del cespite che sono campi non modificabili (*readonly*) e le sei informazioni aggiuntive che possono essere aggiornate (se presenti) o inserite ex novo cliccando sul bottone “Save DATA”.

**Update Asset**

TAG Number

Description

**ADDITIONAL INFO**

Department Responsibility Area

Map Number

Link Photo

Note

Quantity

Last Inventory Date

**Figura 3.7 Le informazioni del cespite selezionato**

Per accedere alle altre funzionalità dell'applicazione è stata creata una pagina html con dei frame, in modo da avere sempre a disposizione sulla parte sinistra dello schermo la possibilità di selezionare l'operazione desiderata.

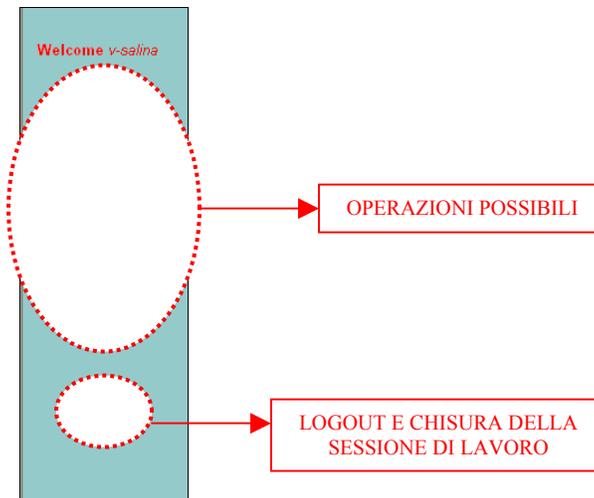


Figura 3.7 Il frame di sinistra

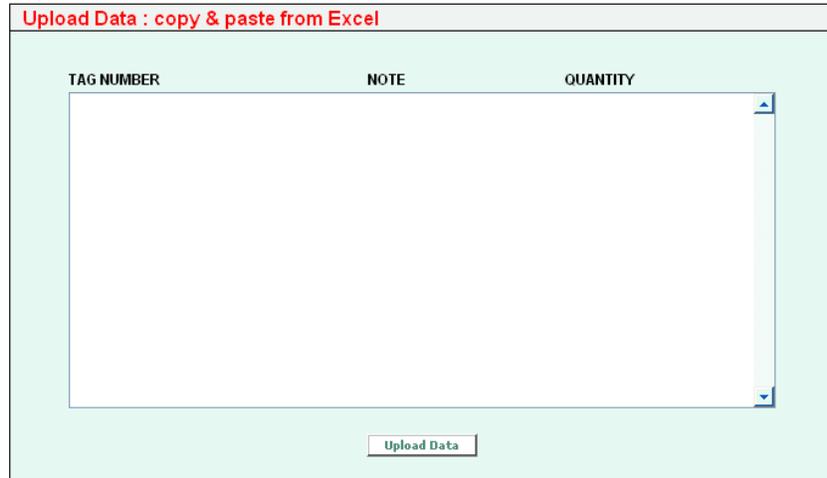
### 3.3.3 Upload DATA

Come detto in analisi di progetto, i dati relativi ai cespiti inventariati vengono ricavati da un foglio excel realizzato appositamente per interagire con l'applicazione web. Questo sheet viene aggiornato ogni volta che lo scanner IR legge il codice a barre mediante delle macro VBA che utilizzano un evento *onChange*. L'utente, una volta che dispone di tale foglio, esegue un copy&paste nella pagina "Upload DATA" dell'applicazione.

Un esempio di come potrebbe presentarsi lo sheet:

Tag Number	Note	Quantity
0126115-000100	Funzionante	1
0126115-000200	Difettato	1
0126115-000300	Funzionanti	2
0126115-000400	Funzionante	1
0126199-000100	Mancante	1

La pagina “Upload DATA” si presenta come segue:



Upload Data : copy & paste from Excel

TAG NUMBER	NOTE	QUANTITY
------------	------	----------

Upload Data

**Figura 3.8 “Upload DATA”**

Effettuato il paste nella textarea e sufficiente cliccare sul bottone “Upload Data” per andare ad aggiornare il database. Viene richiamata la pagina *Exec\_upload\_data.jsp* che esegue tale operazione. Se va a buon fine si ritorna alla pagina di Figura 3.8 che visualizza il messaggio di corretta esecuzione altrimenti viene visualizzata una pagina con il messaggio di errore (Figure 3.9).

La soluzione tecnica adottata per eseguire tale operazione è stata quella di utilizzare uno *StringTokenizer* che è in grado di separare un testo in base al carattere spaziatore, nel nostro caso un tabulatore tra campi della stessa riga (\t) e new line tra righe diverse (\n).



**ATTENTION!! ERROR!!**

You must to paste all and only the columns "TAG\_NUMBER","NOTE","QUANTITY" from Excel

If you can't do it please contact support ✉

Back

**Figura 3.9 Il messaggio di errore**

Capitolo 3. La realizzazione del progetto

---

### 3.3.4 View DATA

Cliccando sul bottone “View DATA” viene aperta una nuova finestra del browser che si collega ad una particolare applicazione chiamata ART (*A Reporting Tool*) e realizzata per gestire la fase di reporting. Questa applicazione permette di eseguire delle query via web, ma anche di esportare il risultato su un foglio excel selezionando l’opportuno *View mode*. Per “lanciare” la query si devono inserire gli opportuni parametri di ricerca come si può vedere da Figura 3.10.

Nella fase di scrittura del codice SQL che genera il report si è dovuto tener conto delle particolarità del caso italiano già descritte nei paragrafi precedenti. Quindi, come nel caso della query per la visualizzazione dei cespiti, si è utilizzato il comando decode per associare correttamente le informazioni relative alle LBC di look-up.

Questa query estrae i dati necessari agli utenti che dovranno eseguire l’inventario fisico dei cespiti, consentendo l’export in fogli excel ai quali saranno applicate le macro per l’aggiornamento o la modifica dei campi consentiti.

Please insert the required parameters for query WEBASSET INFO				
Name	Value	Type	Description	Help
COUNTRY	IT-ITALY		COUNTRY	[...]
TAG_NUMBER	%	ABC	TAG_NUMBER	[...]
LBC	%	ABC	LBC	[...]
BOOK_TYPE_CODE	%_F	ABC	BOOK_TYPE_CODE	[...]
Month	12	1.23	Month	[...]
Year	2004	1.23	Year	[...]
View mode	Spreadsheet (xls)	[...]		Execute

- Browser (Fancy)
- Browser (Plain)
- Spreadsheet (zip slk)
- Spreadsheet (zip xls)
- Spreadsheet (xls)
- Spreadsheet (gzip text tsv)
- Spreadsheet (text tsv)
- Xml

Figura 3.10 La maschera per l’esecuzione della query da ART

Capitolo 3. La realizzazione del progetto

La query implementata è la seguente:

```
SELECT * FROM (
SELECT
A.MONTH,A.YEAR,A.BOOK_TYPE_CODE,A.GL_LEC,A.GL_UEC,
decode(A.LBC|A.ATTRIBUTE13,'09295U126','01266','09295U921','09211',
'09295U9221','09221','09295U9222','09222','09295U924','09243',
'09295U925','09253','09295U927','09273','19295BAREGGIO','19234',
```

### **3.4 Il deployment**

La pubblicazione di un'applicazione web deve seguire delle specifiche che implicano una struttura standard delle directory contenenti pagine HTML, JSP, immagini, classi di supporto e dati di configurazione (J2EE).

La cosiddetta directory radice (*root*) contiene tutte le altre e, solitamente, viene indicata con il nome dell'applicazione, nel nostro caso */AssetApplication*. Al suo interno sono presenti tutte le informazioni che potranno essere offerte direttamente all'utente, a meno della directory speciale denominata */WEB-INF*. Essa contiene file riservati come il file di configurazione *web.xml* (*deployment descriptor*) e le classi e gli archivi Jar di cui l'applicazione web necessita.

Seguendo questi standard di costruzione risulta più semplice effettuare un deployment accelerato utilizzando le modalità del file war. L'applicazione viene conglobata in un archivio *WAR* (*Web Application Archive*) che compatta tutte le sue componenti e permette il trasferimento sul server di un unico file. Nel caso di Tomcat è sufficiente copiare il file nella directory appropriata e riavviarlo; esso provvederà automaticamente a decomprimere l'archivio e ad avviare l'applicazione.

### 3.4.1 La migrazione

La filosofia di Alcoa è quella di fornire ai propri clienti un prodotto affidabile e privo di errori. Per questo l'applicazione web è stata sviluppata in tre ambienti differenti ma con le stesse caratteristiche tecnologiche: sviluppo, test e produzione.

L'ambiente di sviluppo (*development*) è stato utilizzato per far vedere al committente lo stato di avanzamento del progetto e per chiarire i dettagli sulle funzionalità richieste.

Quindi, l'applicazione è passata in ambiente di test (UAT, *User Acceptant Test*) al fine di permettere ad alcuni utenti finali di verificarne il funzionamento.

Infine, dopo l'approvazione del responsabile, sarà effettuata la migrazione in ambiente di produzione, dove l'applicativo verrà effettivamente utilizzato.

## Conclusioni

---

## Conclusioni

L'esperienza di *stage* si è posta come obiettivo la realizzazione di un software che permettesse di rendere più efficace ed affidabile la gestione dei cespiti e del loro inventario. Partendo da una richiesta del responsabile italiano, si è pensato di concretizzare il tutto attraverso l'implementazione di un'applicazione web che rendesse automatico tale processo.

L'applicazione è stata realizzata seguendo le esigenze del committente e verificando continuamente che lo stato di sviluppo rispondesse alle funzionalità richieste, attraverso il confronto diretto, anche con i membri del WRC.

Il lavoro in team, infatti, è un'esigenza prioritaria all'interno di moltissime organizzazioni in quanto la coesione di gruppo aumenta le performance del gruppo stesso, portando al successo di tutti.

Attualmente, l'applicazione si trova in ambiente test (*quality assurance*) in quanto è ancora in fase di test. Si sta verificando che i dati relativi al mese di Gennaio 2005 vengano estratti ed elaborati in maniera corretta.

La migrazione definitiva in ambiente di produzione e, quindi, il passaggio al pieno utilizzo del sistema è previsto a partire dal mese di Febbraio 2005.

L'applicativo sviluppato sarà in grado di rendere automatico il processo di distribuzione dei dati ai vari stabilimenti, eliminando il lavoro manuale da parte del responsabile cespiti. Ogni utente potrà eseguire una query che estrae tutti e solo i dati dei cespiti di cui ha bisogno, per poi passare all'inventario fisico degli stessi ed eseguire l'upload dei cespiti inventariati.

Un limite di performance del sistema sviluppato è rappresentato da una certa lentezza nell'estrazione dei dati. Questo dipende dal gran numero di informazioni presenti nel

---

Conclusioni

database; basti pensare che, solo per l'Italia, sono state inserite circa quindicimila righe nella tabella dei cespiti.

La fase di reporting è da considerarsi uno sviluppo futuro, in quanto i report sono stati sviluppati, per velocità di implementazione e di utilizzo, con uno strumento *web base* piuttosto che con Oracle Discoverer, che è il tool di reporting scelto da Alcoa a livello globale.

Per il momento il sistema sarà utilizzato solamente nel contesto italiano, ma, a breve, verrà distribuito anche a livello europeo, seguendo la filosofia di Alcoa di standardizzazione dei processi informatici.

## Bibliografia

---

## Bibliografia

- [1] **Alcoa Corporation**, <http://www.alcoa.com>, Web Site, 2004
  
- [2] **Alcoa Corporation Annual Report 2003**  
Pittsburgh, Pennsylvania, (USA), 2004
  
- [3] **Il manuale del codice a barre**, <http://www.codiceabarre.it>, Web Site, 2004
  
- [4] Atzeni Ceri Paraboschi Torlone, **“Basi di dati”**, McGraw-Hill, 1999
  
- [5] **Servlet e JSP** <http://www.html.it/jsp/index.html>, Web Site, 2004
  
- [6] Phil Hanna, **“JSP La guida Completa”**, McGraw-Hill, 2001
  
- [7] **Apache Jakarta Tomcat**, <http://jakarta.apache.org/tomcat>, Web Site, 2004
  
- [8] **Apache Jakarta Tomcat**, <http://www.latoserver.it/tomcat>, Web Site, 2004
  
- [9] Groff Weinberg, **“SQL La guida Completa”**, McGraw-Hill, 2003
  
- [10] **Oracle9i SQL Reference Release 2 (9.2)**,  
<http://www.utexas.edu/its/unix/reference/oracledocs>, Web Site, 2004
  
- [11] Oracle Corporation, **PL/SQL User’s Guide and References**, 1995