

POLITECNICO DI MILANO

V Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

Dipartimento di Elettronica e Informazione



TITOLO

**SVILUPPO DI UNA INTERFACCIA GRAFICA BASATA SUL WEB
PER LA LOCALIZZAZIONE DI PAZIENTI**

ANTLab Advanced Network Technologies Laboratory
del Politecnico di Milano

Relatore: Ing. Matteo CESANA

Correlatore: Ing. Alessandro REDONDI

Elaborato di Laurea di:

Igor ROSSA, matricola. N. 711588

Anno Accademico 2010/2011

INDICE GENERALE

Indice figure.....	4
1 INTRODUZIONE.....	7
1.1 ABSTRACT.....	7
1.2 SISTEMI DI MONITORAGGIO.....	8
2 SISTEMA “LAURA”.....	12
2.1 DESCRIZIONE.....	12
2.2 ARCHITETTURA DEL SISTEMA.....	13
2.3 IDENTIFICAZIONE DEI REQUISITI.....	15
2.4 ANALISI PROBLEMATICHE.....	16
3 SOFTWARE UTILIZZATO.....	18
3.1 ADOBE FLASH.....	18
3.2 AJAX.....	20
3.3 MySQL.....	25
4 L'APPLICAZIONE.....	35
4.1 CREAZIONE DELLA BASE DI DATI DI PARTENZA.....	35
4.2 REALIZZAZIONE PORZIONE FLASH.....	38
4.3 REALIZZAZIONE PORZIONE AJAX.....	40
4.4 REALIZZAZIONE PORZIONE JAVA.....	41
4.5 SIMULATORE.....	42
5 USE CASE.....	46
5.1 PAGINA PRINCIPALE PER UTENTI NORMAL.....	46
5.2 PAGINA PRINCIPALE PER UTENTI NORMAL VERSIONE "LINE".....	47
5.3 PAGINA PRINCIPALE PER UTENTI ADMIN.....	48
5.4 PAGINA PRINCIPALE PER UTENTI ADMIN VERSIONE "LINE".....	49
5.5 PAGINE INSERIMENTO DATI CARTELLA.....	50
5.6 PAGINA INSERIMENTO DATI PAZIENTE.....	51
5.7 PAGINE INSERIMENTO DATI SENSORI FISSI.....	52
5.8 PAGINA DI INSERIMENTO NUOVO UTENTE.....	53
5.9 PAGINA PER LA DEFINIZIONE DEI CONFINI DI STANZA.....	54
5.10 PAGINA PER LA DEFINIZIONE DELLA REGOLE DI INTRUSIONE IN UNA STANZA.....	55
6 EVALUATION E PERFORMANCE.....	57
6.1 SIMULAZIONE "BASE".....	58
6.2 SIMULAZIONE "INTERMEDIA".....	61
6.3 SIMULAZIONE "OVER".....	64
7 CONCLUSIONI E SVILUPPI FUTURI.....	69
7.1 CONCLUSIONI.....	69
7.2 SVILUPPI FUTURI.....	70
8 BIBLIOGRAFIA.....	72
9 RINGRAZIAMENTI.....	74
10 APPENDICE A.....	75
10.1 DIAGRAMMI UML.....	75

Indice figure

• Principali componenti architetturali del sistema.....	14
• Rappresentazione di una richiesta ad un server con latenza e risposta.....	22
• Rappresentazione di una richiesta ad un server con latenza e risposta.....	23
• Schema E-R del database.....	36
• Prima parte della pagina web dell'applicazione.....	41
• Seconda parte della pagina web dell'applicazione	41
• Angolo tra punto iniziale e finale.....	43
• Andamento simulazione.....	44
• Pagina principale per utenti Normal.....	46
• Pagina principale utenti Normal versione “Line”	47
• Pagina principale utenti Admin.....	48
• Pagina principale utenti Admin versione “Line”	49
• Pagina dell'applicazione per l'inserimento facilitato dei parametri della cartella paziente.....	50
• Pagina dell'applicazione per l'inserimento facilitato dei parametri del paziente.....	51
• Pagina dell'applicazione per l'inserimento facilitato dei parametri dei sensori fissi.....	52
• Pagina dell'applicazione per l'inserimento facilitato dei parametri dell'utente.....	53
• Pagina dell'applicazione per l'inserimento facilitato delle dimensioni della stanza.....	54
• Pagina dell'applicazione per l'inserimento facilitato delle regole di violazione stanza.....	55
• Andamento dell'applicazione con pochi pazienti.....	58
• Andamento dell'applicazione con pochi pazienti.....	59
• Andamento dell'applicazione con pochi pazienti.....	60
• Andamento applicazione caso intermedio.....	61
• Andamento applicazione caso intermedio.....	62
• Andamento applicazione caso intermedio.....	63
• Andamento applicazione caso over.....	64
• Andamento applicazione caso over.....	65
• Andamento applicazione caso over.....	66
• Pagina BenvenutoAdminLine.....	75
• Pagina BenvenutoAdmin.....	76
• Pagina BenvenutoNormalLine.....	77
• Pagina BenvenutoNormal.....	78
• Pagina Inserimento Stanza.....	79

- **Pagina Inserimento Cartella clinica del Paziente.....80**
- **Pagina Inserimento vincolo di distanza tra pazienti.....80**
- **Pagina di Inserimento Stanza da Evitare.....81**
- **Pagina Inserimento dati Paziente.....81**
- **Pagina Inserimento dati Sensore.....81**
- **Pagina Inserimento dati Utente.....82**

1 Introduzione

1.1 Abstract

La realizzazione di un'applicazione di INDOOR POSITIONING, per la visualizzazione della posizione di persone (degenti) all'interno di una struttura (“casa di cura”) fa parte di un progetto su scala più ampia (PROGETTO LAURA) che si occupa dell'analisi e della realizzazione dell'intero e complesso sistema di rilevazione, conversione e visualizzazione di tutti i dati relativi al Paziente-Degente. Questo Lavoro è particolarmente vasto, dato che riguarda un numero molto elevato di discipline (informatica, Elettronica, Telecomunicazioni, ...), e questa sua vastità è anche dovuta al fatto che le informazioni da reperire e visualizzare provengono da una elevata quantità di fonti differenti.

Queste fonti, per essere utilizzate al meglio, hanno necessariamente richiesto un periodo più o meno lungo di “apprendistato” e più in generale, di “adattamento” per poter essere sfruttare appieno.

Si è pensato alla realizzazione di un'interfaccia Web per tale Lavoro, in quanto risulta essere il modo più efficiente ed efficace di raggiungere una grande quantità di utenti non necessariamente dotati di particolari “Skill”, ma semplicemente dei normali “Internauti”. Il Web è ormai riconosciuto “de-facto” come il modo più economico e veloce per pubblicizzare, diffondere ed usufruire di informazioni, di svariato tipo e natura, ed è per questo che la nostra scelta è ricaduta su questa tecnologia.

Per immagazzinare e analizzare le informazioni/dati che l'applicazione fornisce si è quindi dovuto ricorrere all'uso di database, nello specifico di MySQL, una tecnologia che permette di risolvere query di analisi ed elaborazione. Queste query, sono state ottimizzate per il database in considerazione in modo da limitare il carico del DBMS (DataBase Management System), in modo da prevenire il sovraccarico ed eventuale blocco degli applicativi.

A questo punto è sorto il problema della realizzazione di una base di dati che permettesse di ospitare tutti i dati e consentisse il facile reperimento delle informazioni più importanti riguardanti il Paziente e la sua posizione all'interno della Struttura. Dopo aver scelto la

tecnologia per il database ci siamo dovuti subito chiedere quale fosse quella migliore per realizzare la nostra applicazione Web. La nostra attenzione è caduta su Java ed in particolare sulle sua “appendice Web”, cioè JSP (Java Server Pages) e Servlet. Volendo, però, sfruttare al massimo la nuove tecnologie che il Mondo del Web mette a disposizione, non ci siamo limitati a JSP e Servlet, bensì abbiamo intrapreso la nuovissima strada denominata “AJAX”. Per concludere abbiamo optato per un applicativo Flash (vers.8 actionscript 2.0) che realizzasse la mappatura vera e propria di tutti i Pazienti all'interno della Struttura e che controllasse tutte le regole in vigore su di essi. L'ambito del nostro lavoro è quindi stato: realizzare un database, un programma per inserire i dati nel DB in modo facilitato, rapido e veloce, realizzare un applicativo Flash che renderizzasse in modo chiaro, rapido ed aggiornato la posizione e lo “stato” in cui si trova OGNI Paziente. Sempre in Flash è stata fornita anche una versione che renderizzasse, non la posizione e lo stato dei pazienti, ma le regole che essi violano (renderizzato tramite “log testuale”) . “Last but not least” abbiamo realizzato una porzione di codice atto ad eseguire alcune modifiche (prevalentemente di impostazione) di dati e proprietà dei vari pazienti SENZA perderli di vista (in pratica nella stessa pagina si possono modificare dei parametri del database e allo stesso tempo avere la visione dell'applicazione di posizionamento, in modo da avere sempre la situazione sotto controllo). L'obiettivo principale del lavoro era poter usufruire di tutte le funzioni e dei dati offerti dal database senza distogliere lo sguardo dall'applicazione principale (quella di INDOOR POSITIONING); questo vincolo, fondamentale per il corretto utilizzo, ha creato un po' di difficoltà in quanto per risolverlo ci si è dovuti dedicare allo studio di flash, ma anche e soprattutto di AJAX.

Diverse sono state le problematiche sorte anche durante la realizzazione della base di dati, tra le quali la scelta di tabelle e campi che potessero racchiudere al meglio tutte le informazioni, sensibili e non, la realizzazione di un'interfaccia grafica in grado di proporre in modo facilitato i comandi per il popolamento della base di dati.

1.2 Sistemi di monitoraggio

Un recente studio riferisce che il 40% delle morti improvvise negli ospedali si verifica nei reparti di medicina generale. Circa un terzo di tutte le morti avviene subito dopo il

trasferimento del paziente dal reparto di terapia intensiva a quello di medicina generale, non solo, ma secondo uno studio dell'American National Patient Safety Agency, l'11% di tutte le morti è la conseguenza di un repentino peggioramento dello stato di salute del paziente ospedalizzato che non viene riconosciuto in tempo o viene trattato in maniera non opportuna. Per queste ragioni è sempre più necessario realizzare progettati per facilitare il personale di assistenza nella valutazione della misura dei parametri vitali dei pazienti.

Per quanto riguarda il monitoraggio dei parametri vitali, il 70% dei pazienti colpiti da un peggioramento delle condizioni cardiache e della funzionalità del sistema respiratorio, manifesta un'acutizzazione nelle otto ore successive all'evento. La registrazione e la documentazione dei principali parametri vitali risulta attualmente insufficiente in molti ospedali.

Le soluzioni che si stanno presentando sul mercato rappresentano una svolta nella rilevazione dei parametri vitali dei pazienti e aiuteranno a semplificare le procedure cliniche migliorando la qualità dell'assistenza. Inoltre la crescente richiesta del mercato per quel che riguarda la rilevazione dei parametri vitali in contesti ospedalieri a bassa criticità rappresenta un'importante area di sviluppo nel miglioramento della sanità mondiale. Considerato il trend di invecchiamento della popolazione mondiale, il personale medico ospedaliero si dovrà confrontare con pazienti sempre più anziani e con malattie croniche. Dato che le unità di terapia intensiva (ICUs) stanno diventando sempre di più una risorsa limitata, i pazienti saranno presto spostati dalle ICUs ai reparti di medicina generale. Allo stesso tempo, negli ospedali sta anche aumentando il focus sulla prevenzione degli errori medici e degli eventi imprevisti: un metodo per garantire una stretta supervisione è il frequente monitoraggio dei parametri vitali corporei dei pazienti e la creazione di una notifica automatica delle anomalie. Oggi, tuttavia, solo il 40% dei letti ospedalieri è dotato di un sistema di monitoraggio del paziente capace di operare in questo modo.

In ambito di assistenza sanitaria è richiesto il frequente monitoraggio di molti pazienti fuori dai reparti di terapia intensiva, per fare ciò ci si deve basare su “manuali delle procedure di monitoraggio” che sono spesso poco esaurienti sebbene i segnali di pericolo si presentino tipicamente nelle fasi iniziali. I cambiamenti improvvisi nello stato di salute dei pazienti non sono sempre rilevati in tempo, quindi un monitoraggio intelligente e un tempestivo sistema

di notifica può aiutare ad intervenire fin dai primi momenti per evitare il verificarsi di eventi imprevisti, aumentando quindi considerevolmente la sicurezza dei pazienti. Il sistema di monitoraggio verrà introdotto a partire dal prossimo anno in America, Scandinavia, Regno Unito e Australia, paesi in cui è già stato istituito il così detto “Rapid Response Team”, ossia la creazione di un gruppo di specialisti che interviene rapidamente sull’aggravarsi delle condizioni dei pazienti. Studi scientifici hanno dimostrato che ricorrendo a tale sistema si può considerevolmente ridurre il tasso di mortalità e morbilità (di circa il 33.8%).

2 Sistema “LAURA”

2.1 Descrizione

Negli ospedali e nelle case di cura con pazienti che soffrono di disturbi cognitivi e/o fisici, uno dei problemi più comuni e significativi è l'efficienza dell'attività di controllo e supervisione dei pazienti. Questo problema può essere suddiviso in due aspetti principali:

- **localizzazione:** il personale medico deve essere in grado di sapere in ogni momento l'esatta posizione di ciascun paziente deambulante; il valore di precisione di tale dato può variare in base a diversi fattori.
- **Monitoraggio dello stato:** il personale medico deve essere sempre a conoscenza dello stato aggiornato dei principali livelli biometrici di ogni paziente. Si monitorano diversi parametri in modo da rilevare in essi una variazione che può rappresentare una situazione di pericolo.

Ad oggi alcuni tentativi di sviluppare un sistema capace di adempiere a tali problematiche si sono concentrati sulle tecnologie GPS e video. Tali scelte si sono tuttavia rivelate inadeguate, in quanto soluzioni basate sul sistema GPS incontrano grosse difficoltà quando la localizzazione deve avvenire in luoghi chiusi, mentre soluzioni che prevedono l'utilizzo di videocamere non svincolano il personale medico dalla necessità di osservare continuamente i soggetti.

Considerando che esistono dispositivi mobili in grado di monitorare valori biometrici critici o di raccogliere dati dai quali si è in grado di dedurre alcune situazioni di pericolo, è nata l'idea per una nuova soluzione: l'utilizzo di apparecchiature “indossabili” dai soggetti monitorati in una rete di sensori, per permettere l'analisi centralizzata dei segnali da essi generati. Grazie a questa centralizzazione, l'attività di localizzazione e gran parte delle attività di monitoraggio possono essere completamente delegate ad un sistema automatizzato. Una soluzione di questo tipo deve essere in grado di rispondere ai seguenti requisiti:

- I dispositivi applicati ai pazienti devono essere quanto meno invasivi possibile.
- I dispositivi devono periodicamente comunicare con un sistema centrale per la

trasmissione dei dati raccolti.

- I sensori devono avere la capacità di comunicare in modo wireless.
- I sensori devono avere un'adeguata autonomia.
- Il sistema deve funzionare nell'intera area di interesse, sia essa un ambiente chiuso, che uno spazio aperto.
- Il sistema deve essere di facile installazione, fruizione ed il più compatibile possibile con le strutture preesistenti.

E' proprio nel tentativo di concretizzare questa idea innovativa che il progetto LAURA (LocAlization and Ubiquitous monitoRing of pAtients for healthcare support) si cimenta. La sperimentazione di tale sistema avviene in un contesto di cooperazione con la "Fondazione Eleonora e Lidia" di Figino Serenza (CO), un piccolo istituto specializzato nell'assistenza a persone con particolari necessità. La sperimentazione verterà principalmente sulla localizzazione e sul monitoraggio di pazienti che soffrono di attacchi epilettici, al fine di garantire un tempestivo intervento.

2.2 Architettura del sistema

In questa sezione valuteremo molto brevemente le principali componenti necessarie al funzionamento del sistema descritto nell'introduzione. Il sistema può essere in prima analisi suddiviso in tre aree principali.

- **Personal Monitoring System (PMS):** si occupa del monitoraggio centralizzato dello stato di ciascun paziente attraverso l'uso dei dispositivi wireless precedentemente citati. Tali dispositivi, indossati dai pazienti, non ne ostacoleranno le normali attività. Nel caso specifico della nostra cooperazione con la "Fondazione Eleonora e Lidia" ci limiteremo all'impiego di accelerometri al fine di rilevare eventuali cadute o crisi epilettiche dei degenti. Il PMS è composto da un'architettura di rete leggera e non intrusiva oltre che da una serie di algoritmi di elaborazione dei segnali in grado di ottimizzare l'informazione rilevata rispetto l'architettura scelta.
- **Personal Localization System (PLS):** si occupa, come suggerisce il nome, della localizzazione dei pazienti all'interno della rete di sensori, al fine di garantire un

tempestivo intervento in caso di necessità. In base alla tecnologia utilizzata tale localizzazione può avvenire in modo più o meno accurato; avremo dati più veritieri se ci basiamo su ricevitori di ultrasuoni, mentre saremo meno precisi se ci si basa sulla misurazione dell'energia associata ai segnali emessi dai vari sensori. Così come il PMS anche il PLS è fondamentalmente composto da un'architettura di rete particolarmente “leggera” e da algoritmi di elaborazione di segnali, che vengono maggiormente centralizzati al fine di aumentare l'autonomia dei nodi, riducendone il carico di lavoro.

- Network Architecture (NA): implementa algoritmi di routing ad-hoc per una rete complessa come quella descritta, minimizzando il consumo energetico dei singoli nodi.

Nella figura, viene illustrata l'architettura con le sue componenti principali (non tutte le componenti illustrate in questa figura sono state commentate nella descrizione precedente).

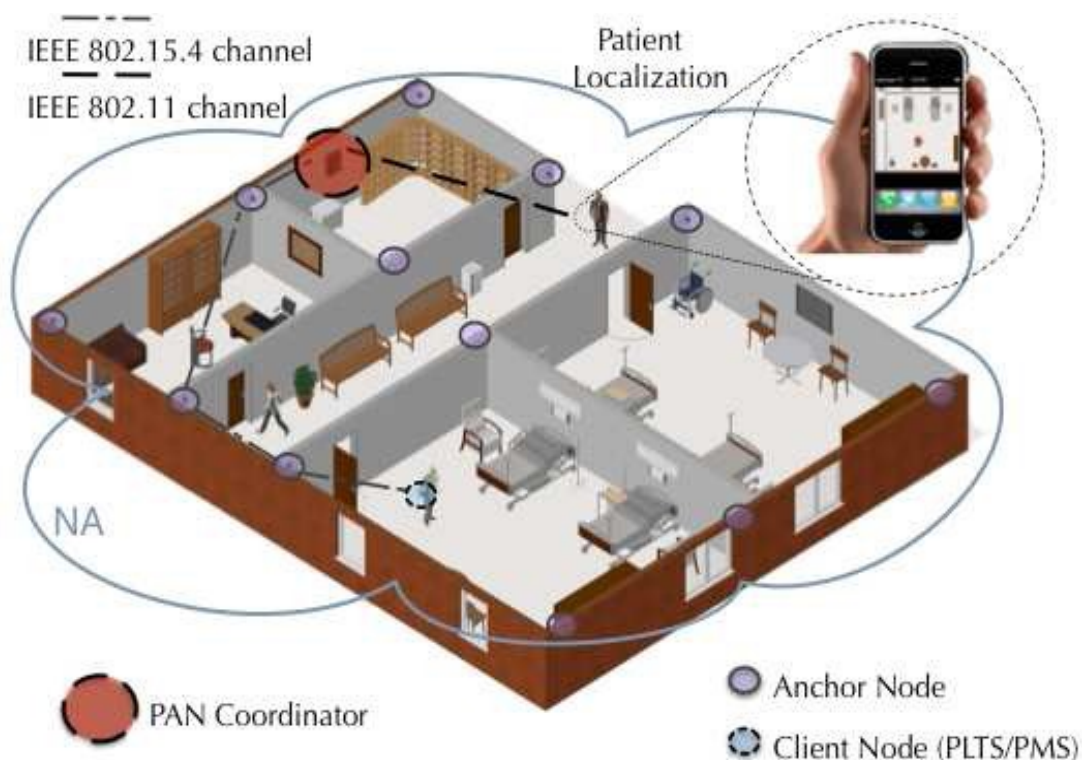


Illustrazione 2.1: Principali componenti architetture del sistema

Non è utile, in questa trattazione, addentrarsi in ulteriori dettagli sull'architettura di LAURA, in quanto le informazioni date finora sono sufficienti alla comprensione dell'elaborato.

2.3 Identificazione dei requisiti

Valutiamo ora lo scopo principale di questo trattato e le principali problematiche associate al suo conseguimento. Bisogna innanzitutto specificare in quale porzione del complesso sistema descritto sinora si colloca l'attività; il risultato della ricerca sarà il software di monitoraggio centrale per il sistema LAURA, l'unico punto di contatto del sistema con il personale medico. Esso ricopre un ruolo fondamentale, in quanto permette di sfruttare al meglio le potenzialità di LAURA, fornendo uno strumento ricco di funzionalità di alto livello volte al conseguimento degli obiettivi principali del progetto. Utilizzando la semplice informazione di localizzazione, saremo in grado di rilevare in maniera automatizzata il verificarsi di situazioni di potenziale pericolo prima che esse si concretizzino; sarà ad esempio possibile istruire il sistema in modo che segnali la presenza di un soggetto monitorato in un ambiente ad esso non adatto.

Per garantire che il software si adatti nel migliore dei modi al ruolo che ricopre all'interno del progetto, si rende necessario che i seguenti requisiti vengano soddisfatti:

- Immediatezza della consultazione dei dati: nel contesto descritto i dati generati dal sistema sono numerosi e complessi, è quindi di fondamentale importanza presentarli nella maniera più semplice, completa e soprattutto di immediata interpretazione possibile.
- Semplicità di gestione ed utilizzo: sarà di notevole importanza garantire la massima autonomia di gestione al personale medico, senza la richiesta di alcuna competenza specifica per lo svolgimento delle normali attività di utilizzo ed aggiornamento del sistema.
- Stabilità del sistema: per via della sua funzione, l'applicazione deve garantire la massima affidabilità anche in condizioni non ideali e di sovraccarico.

2.4 Analisi problematiche

Vediamo ora le scelte effettuate per ottemperare ai requisiti definiti nella sezione precedente:

- Immediatezza della consultazione dei dati: per soddisfare questo requisito si è scelto di implementare un'interfaccia in cui buona parte dello spazio disponibile è occupato da una rappresentazione grafica della planimetria dell'edificio. Su di essa verranno visualizzate in tempo reale icone in corrispondenza delle posizioni occupate dai pazienti monitorati, in questo modo con una rapida osservazione sarà possibile conoscere la posizione di ogni soggetto. La segnalazione di situazioni di pericolo avviene tramite “faccine” colorate che oltre ad essere indicatrici della posizione del paziente sono anche in grado di attirare immediatamente l'attenzione del personale addetto. Lo “smile” del paziente sarà verde se tutto è ok, giallo se si sta per verificare un pericolo e rosso se il paziente è in uno stato di immediato pericolo. Tutto questo risulterà facilmente visibile sulla mappa facente parte dell'applicazione.
- Semplicità di gestione ed utilizzo: sono stati presi vari accorgimenti per semplificare il più possibile l'utilizzo del software da parte del personale medico, ne presenteremo alcuni tra i principali. All'avvio del software non è necessario svolgere alcuna particolare attività, dato che il sistema è in grado di inicializzarsi in maniera autonoma, ripristinando lo stato del sistema al momento dell'ultimo utilizzo. Anche in fase di monitoraggio le interazioni con il software da parte dell'utente sono elementari, riguardano infatti semplicemente la gestione degli allarmi che vengono notificati. Le operazioni di inicializzazione e/o aggiornamento dei dati utili al sistema sono state rese il più possibile guidate ed intuitive, realizzando interfacce in grado ad esempio di notificare eventuali inconsistenze dei dati prima della loro immissione nel programma.
- Stabilità del sistema: questo requisito è stato soddisfatto prediligendo in generale un codice consistente e meno esposto ad errori a run-time, rispetto ad uno più performante, ma meno solido.

3 Software utilizzato

3.1 Adobe Flash

Adobe Flash (in precedenza Macromedia Flash e ancora prima FutureSplash) è un software per uso prevalentemente grafico che consente di creare animazioni vettoriali principalmente per il web. Viene utilizzato inoltre per creare giochi o interi siti web e grazie all'evoluzione delle ultime versioni è divenuto un potente strumento per la creazione di “Rich Internet Applications” e piattaforme di streaming audio/video. Viene inoltre incorporato nei Media Internet Tablet (M.I.T.) della Archos.

Caratteristiche:

Flash permette di creare animazioni complesse e multimediali. All'interno di esse infatti si possono inserire:

- forme vettoriali, che sono gli oggetti principali con cui Flash permette di lavorare.
- Testo (sia statico che dinamico) e caselle di input per il testo.
- Immagini raster (Bitmap, GIF, Jpeg, PNG, TIFF e altri formati) sottoforma di oggetto bitmap.
- Audio (MP3, WAV e altri), sia in streaming che per effetti sonori.
- Video (AVI, QuickTime, MPEG, Windows Media Video, FLV).
- Altre animazioni create con Flash (tramite Actionscript o interpolazioni).

Inoltre permette di creare animazioni interattive, grazie alla presenza di un linguaggio di scripting interno. Tramite questo linguaggio, denominato ActionScript e basato su ECMAScript, è possibile applicare comportamenti agli oggetti o ai fotogrammi dell'animazione. Inizialmente erano presenti solo poche azioni, ma allo stato attuale

ActionScript è diventato uno strumento molto potente. Grazie ad esso si possono creare infatti menù, sistemi di navigazione, GUI, siti web completi e giochi anche complessi.

Un'altra caratteristica importante è la possibilità di mantenere **simboli** riutilizzabili in una **libreria**. Esistono diverse tipologie di simboli, fra i quali semplici oggetti grafici, **clip filmato**(sotto-animazioni con una timeline propria e indipendente da quella principale), pulsanti, oggetti bitmap, ecc. Questa caratteristica è molto importante, in quanto permette di:

- semplificare il lavoro dello sviluppatore, che può modificare tutte le istanze presenti nell'animazione modificando il simbolo nella libreria.
- Ridurre le dimensioni del file SWF prodotto.
- Creare comportamenti interattivi complessi applicando del codice agli oggetti.

Flash Player:

Per visualizzare i siti e le applicazioni creati con flash e Flex, Adobe mette a disposizione gratuitamente il player.

L'ultima versione rilasciata è la 10.3.181.14, disponibile per Microsoft Windows_98, Me, 2000, XP, Server 2003, Mac OS X, Linux_e Android OS 2.2 Froyo.

Il player è disponibile in versione stand-alone_e come plugin per i browser(o controllo ActiveX_per Internet Explorer).

Lo stesso player è disponibile per altre piattaforme_quali Pocket PC, Maemo, OS/2, Sun Solaris, HP-UX, IRIX. Anche la Playstation 3 e la console casalinga Nintendo Wii, supportano la visualizzazione del filmati SWF, la prima tramite un plug-in proprietario (alla versione 7), la seconda tramite il plug-in per il browser integrato(alla versione 9), la terza anch'essa tramite il plug-in del browser Opera integrato.

È disponibile anche una versione del Flash player Macromedia per dispositivi mobili, denominatoMacromedia Flash Lite e giunto alla versione 3, che permette la visualizzazione di filmati flash SWF direttamente su cellulari e altri device.

Il principale software libero per vedere animazioni Flash è Gnash.

Nel mondo Internet il formato Flash (estensione .swf) degli oggetti creati con l'omonimo programma, rappresentano ormai uno standard per la creazione di contenuti animati ed interattivi.

Ormai la quasi totalità dei browser supportano il plug-in del player che consente la visione, appunto, di animazioni di grafica vettoriale così come la visione in streaming di filmati video. Windows XP, ad esempio, include di default la versione 5 del controllo ActiveX per Internet Explorer.

Rischio per la privacy:

Flash Player può essere un rischio per la privacy dell'utilizzatore poiché può immagazzinare sul suo hard disk file .sol che funzionano come cookie, e che potrebbero non essere riconosciuti dai più comuni software anti-malware qualora il loro uso sia maligno. Per proteggersi si possono cercare i file .sol con le funzioni di ricerca, e poi cancellarli manualmente, ma è necessario conoscere il loro uso attuale per evitare inconvenienti. È comunque possibile modificare (ed eventualmente disattivare) questa caratteristica dal Pannello Impostazioni generali della memorizzazione di Flash Player.

È importante sapere che dopo aver disattivato questa funzione non sarà possibile utilizzare applicativi flash che memorizzano risorse localmente, come alcuni giochi online, se l'autore degli stessi non ha previsto questa evenienza.

3.2 AJAX

Introduzione:

Si sente molto parlare di AJAX e non sempre in modo chiaro. L'esigenza di avere una guida chiarificatrice sull'argomento è diventata sempre più reale al punto da far nascere siti

interamente dedicati, ma che partono il più delle volte dal presupposto che chi legge sappia già molto di JavaScript o dello stesso AJAX. Questa guida vorrebbe quindi far luce sulla questione, spiegando dettagliatamente l'acronimo tanto mormorato e le sue potenzialità al fine di fornire ai meno aggiornati o meno preparati su Javascript, le conoscenze per usare questa tecnica di sviluppo.

Cenni storici:

L'acronimo AJAX, che significa esattamente Asynchronous JavaScript And XML (JavaScript asincrono ed XML), è stato enunciato per la prima volta da Jesse Garrett, nel 18 Febbraio 2005, come titolo di un post all'interno del suo blog.

Non si tratta di una nuova tecnologia né di un'invenzione bensì di un concetto utilizzato per sviluppare applicativi avanzati e particolari quali Gmail, Google Maps o Google Suggest. Il concetto è in parte espresso nell'acronimo scelto, un utilizzo asincrono di Javascript che attraverso l'interfacciamento con XML, può permettere ad un client di richiamare informazioni lato server in modo veloce e trasparente, allargando gli orizzonti delle **rich internet applications**. Queste applicazioni fino a poco tempo fa erano legate principalmente alle tecnologie Adobe-Macromedia Flash o Java (con le applet). Entrambe purtroppo non sempre interpretabili dai client degli utenti e troppo spesso usate a sproposito con il solo scopo di stupire, discorso che spesso e purtroppo vale anche oggi. In alternativa a queste tecniche di interazione client/server, quando nel 1996 venne introdotto l'iframe in Internet Explorer 3, molti sviluppatori sfruttarono quest'ultimo modificando l'attributo sorgente (src) della pagina racchiusa e simulando così un refresh trasparente di una parte di contenuti il che emulava, in modo abbastanza sporco, un'interazione asincrona.

Nel 1998 Microsoft cominciò a sviluppare una tecnologia, chiamata Remote Scripting, con lo scopo di creare una tecnica più elegante per richiamare contenuti differenti ed è in questo periodo, seppur con nome differente, che AJAX venne utilizzato per la prima volta, per poi evolversi in versioni più mature fino a diventare un oggetto vero e proprio, noto ora come XMLHttpRequest.

Il motivo principale di tanto successo è che solo ultimamente il Remote Scripting ha suscitato lo stupore degli addetti ai lavori nel vedere cosa Google fosse riuscita a fare all'interno dei suoi applicativi senza necessità di Flash Player o Java Virtual Machine,

mantenendo comunque la compatibilità con molteplici browser di utenti che per diversi motivi non potevano usufruire di Javascript. Come spesso è accaduto negli ultimi anni, molti hanno preso spunto da Google ed il web, noto per la sua immediatezza propositiva, è stato in grado in poco più di un anno di regalarci numerosi esempi di applicativi basati su AJAX, esagerando in alcuni casi nell'utilizzo ma considerando molto spesso e fin da subito, a differenza di quanto accadde per Flash ed Applets anni prima, due delle problematiche più diffuse del web attuale: accessibilità ed usabilità.

Descrizione tecnica e teorica:

Se parliamo di AJAX, oggi, parliamo di un oggetto specifico: XMLHttpRequest. A seconda del browser usato prende nomi differenti o viene richiamato in maniera differente.

Nel caso di Internet Explorer, ad esempio, questo oggetto è restituito da un ActiveXObject mentre nei browser alternativi più diffusi (Mozilla, Safari, FireFox, Netscape, Opera ed altri) XMLHttpRequest è supportato nativamente, cosa che dovrebbe accadere anche per IE dalla versione 7. Questo oggetto permette di effettuare la richiesta di una risorsa (con HTTP) ad un server web in modo indipendente dal browser. Nella richiesta è possibile inviare informazioni, ove opportuno, sotto forma di variabili di tipo GET o di tipo POST in maniera simile all'invio dati di un form.

La richiesta è asincrona, il che significa che non bisogna necessariamente attendere che sia stata ultimata per effettuare altre operazioni, stravolgendo sotto diversi punti di vista il flusso dati tipico di una pagina web. Generalmente infatti il flusso è racchiuso in due passaggi alla volta, richiesta dell'utente (link, form o refresh) e risposta da parte del server per poi passare, eventualmente, alla nuova richiesta da parte dell'utente.

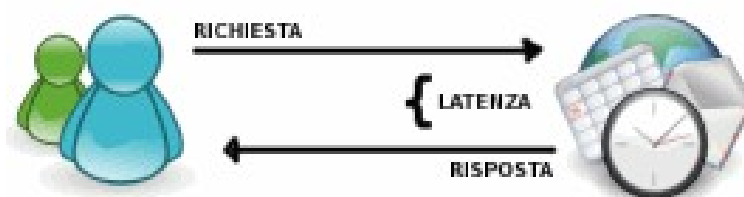


Illustrazione 3.1: Rappresentazione di una richiesta ad un server con latenza e risposta

Il terzo ed inevitabile incomodo di questo ciclo è l'attesa che trascorre tra una richiesta dell'utente e la risposta del server. Con l'aggiunta di AJAX si perde questa linearità e mentre l'utente è all'interno della stessa pagina le richieste sul server possono essere numerose e completamente indipendenti. Nulla infatti vieta, teoricamente, di effettuare decine di richieste simultanee al server per operazioni differenti con o senza controllo da parte del navigatore.

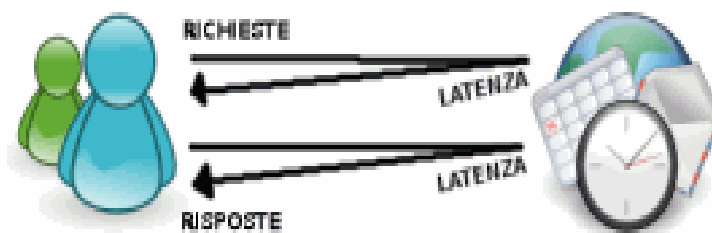


Illustrazione 3.2: Rappresentazione di una richiesta ad un server con latenza e risposta

Ciò che resta del vecchio flusso, il tempo di attesa, passa spesso in secondo piano ed in molti tipi di interazione è praticamente impercettibile. Ma attenzione poiché questo tempo è anche uno dei maggiori problemi dell'utilizzo di AJAX, sia per gli sviluppatori sia per i navigatori. I primi potrebbero trovarsi in difficoltà qualora l'operazione asincrona dovesse attendere assolutamente una risposta al fine di completare una serie di operazioni più o meno sensibili mentre i secondi potrebbero non avere idea di cosa stia accadendo alla pagina chiudendola ignari di aver richiesto un'informazione. A completare questa breve introduzione tecnica e teorica è il tipo di risposta che l'oggetto si aspetta dopo una chiamata che non deve essere necessariamente di tipo XML o letta come tale ma che può essere semplicemente testuale, in contro tendenza con l'acronimo stesso ma non per questo inusuale.

Assegnazione cross-browser:

Ciò che contraddistingue Internet Explorer dagli altri browser è l'assegnazione dell'oggetto XMLHttpRequest. Non solo è parte di un ActiveX e quindi esposto maggiormente a

restrizioni sulla configurazione del browser dell'utente, ma è anche supportato, seppur malamente, dalla versione 4.

Volendo evitare di mostrare errori a chi naviga con questa vecchia versione è necessario verificare che la stringa MSIE 4 non sia presente nel nome che contraddistingue il browser, ovvero navigator.userAgent, che per sicurezza ancora maggiore viene assegnato ad una variabile dopo essere stato cambiato nel corrispettivo in maiuscolo.

Prima ancora è necessario controllare che il supporto all'oggetto XMLHttpRequest sia nativo o che il browser sia almeno di tipo Internet Explorer.

Dichiarazione dell'oggetto XMLHttpRequest cross-browser

```
// FUNZIONE PER ASSEGNARE L'OGGETTO XMLHttpRequest
// COMPATIBILE CON I BROWSERS PIÙ RECENTI E DIFFUSI
FUNCTION ASSEGNAXMLHttpRequest() {

// LISTA DELLE VARIABILI LOCALI
VAR
// VARIABILE DI RITORNO, NULLA DI DEFAULT
XHR = NULL,

// INFORMAZIONI SUL NOME DEL BROWSER
BROWSERUTENTE = NAVIGATOR.USERAGENT.ToupperCase();

// BROWSER STANDARD CON SUPPORTO NATIVO
// NON IMPORTA IL TIPO DI BROWSER
IF(TYPEOF(XMLHttpRequest) === "FUNCTION" || TYPEOF(XMLHttpRequest) === "OBJECT")
  XHR = NEW XMLHttpRequest();

// BROWSER INTERNET EXPLORER
// È NECESSARIO FILTRARE LA VERSIONE 4
ELSE IF(
  WINDOW.ActiveXObject &&
  BROWSERUTENTE.INDEXOF("MSIE 4") < 0
){

// LA VERSIONE 6 DI IE HA UN NOME DIFFERENTE
// PER IL TIPO DI OGGETTO ActiveX
IF(BROWSERUTENTE.INDEXOF("MSIE 5") < 0)
  XHR = NEW ActiveXObject("Msxml2.XMLHTTP");

// LE VERSIONI 5 E 5.5 INVECE SFRUTTANO LO STESSO NOME
```



```
ELSE
  XHR = NEW ActiveXObject("Microsoft.XMLHTTP");
}

RETURN XHR;
}
```

La funzione mostrata è solo una delle tante presenti in rete per istanziare un oggetto XMLHttpRequest a seconda del browser e della sua versione ma è una delle poche che raramente potrà fallire o generare errori anche su computer non ancora aggiornati.

Il concetto alla base è tentare di filtrare i browser riconosciuti come validi per l'utilizzo di AJAX e sfruttare l'eventuale ritorno nullo a proprio favore, inizializzando l'intero applicativo AJAX solo se l'oggetto è riconosciuto come valido.

Un esempio di quanto appena detto potrebbe essere questo:

Riconoscere un browser abilitato ad AJAX

```
ONLOAD = FUNCTION() {
  VAR AJAX = ASSEGNAXMLHTTPREQUEST();
  IF(AJAX) {
    // APPLICATIVO AJAX
  }
}
```

dove qualora l'oggetto non dovesse essere stato assegnato o dovesse essere ancora nullo o false, non si disturberebbe l'utente arrivato con messaggi di errore o con la comparsa del triangolo giallo solitamente mostrato ogni qualvolta siano presenti dei problemi nella pagina visualizzata.

3.3 MySQL

Introduzione:

MySQL, il più popolare database management system open source, è sviluppato, distribuito e

supportato da MySQL AB. MySQL AB è una compagnia commerciale fondata dagli svilupparoti di MySQL.

MySQL è un database management system; un database è una collezione strutturata di dati. Tali dati possono variare dalla lista della spesa ad una fotogallery fino a contenere i dati di una grande azienda. Per aggiungere, accedere e processare i dati immagazzinati in un data base si ha la necessità di avere un database management system come ad esempio MySQL Server.

MySQL è un database management system relazionale; un database relazionale immagazzina i dati in tabelle separate e solo in un secondo momento mette tutti i dati in un'unica area denominata storeroom. Questa particolarità conferisce velocità e flessibilità. La denominazione SQL di "MySQL" significa "Structure query language". SQL è il più comune e standardizzato linguaggio per l'accesso a base di dati. Lo standard SQL si è evoluto fin dal 1986 e ad oggi ne esistono molte versioni. Per esempio "SQL-92" è riferito allo standard rilasciato nel 1992,"SQL:1999" si riferisce allo standard rilasciato nel 1999 e "SQL:2003" è riferito alla attuale versione dello standard.

MySQL è un software Open Source; Open Source significa che è possibile a chiunque usare e modificare il software. Chiunque può scaricare il software di MySQL da Internet e utilizzarlo senza pagare nulla. Se si ha la necessità si può studiare il codice sorgente per poterlo adattare al meglio alle proprie esigenze. Il software MySQL usa la GPL (GNU general public license), per definire ciò che si può e ciò che non si può fare con il software in diverse occasioni.

MySQL database Server è molto veloce, riutilizzabile e facile da usare; inoltre MySQL ha anche un ampio e pratico insieme di comportamenti utile a chiunque, dal principiante al programmatore più esperto. MySQL Server fu originariamente creato per manipolare grandi database più veloci delle esistenti soluzioni e fu utilizzato con successo in innumerevoli ambienti di produzione per molti anni. Attraverso un costante sviluppo, MySQL oggi offre un ricco e utile insieme di funzioni. La sua connettività, velocità e sicurezza l'ha reso molto adatto all'accessibilità ai database tramite Internet.

MySQL lavora in sistemi client/server o embedded; il software MySQL è un sistema client/server che consiste in un SQL Server multi-threaded che supporta diversi tipi di

backend, librerie e programmi client, ecc...

MySQL Server è supportate da tutte le maggiori applicazioni e linguaggi di programmazione.

La storia di MySQL:

L'intenzione primaria dell'uso di mSQL era quella di collegare alle nostre tabelle le routine di basso livello (ISAM). Comunque dopo vari test, si era giunti alla conclusione che mSQL non era abbastanza veloce e flessibile per tali scopi. Questo sfociò nella realizzazione di una nuova interfaccia SQL, la quale però conteneva tutte le API dell'interfaccia mSQL. Queste API furono realizzate per permettere a codice di terze parti scritto per mSQL di poter essere facilmente utilizzato anche per MySQL.

L'origine del nome MySQL non sono chiare. La maggior parte dei file e delle cartelle avevano avuto per svariati anni come prefisso "my".....comunque anche la figlia del fondatore si chiamava My. Quale delle due diede il nome a MySQL è tuttora un mistero.

Le caratteristiche di MySQL:

La seguente lista descrive alcune delle più importanti caratteristiche di MySQL.

Caratteristiche realizzative e portabilità:

- Scritto in C e C++.
- Testato con differenti compiler.
- Lavora su diversi tipi di piattaforme.
- Usa GNU Automake, Autoconf e Libtool per la portabilità.
- Sono disponibili API per C, C++, Eiffel, Java, Perl, PHP, Python, Ruby e Tcl.
- Può facilmente usare CPU multi-core.
- Ha un veloce sistema di allocazione della memoria basato su thread.

- Realizza join molto veloci basandosi sulla tecnica denominata “una passata multi-join”.

Le funzioni SQL sono implementate usando una class library altamente ottimizzata (la più veloce che sia possibile).

Il server è disponibile come un programma separato per essere usato in un ambiente client/server.

Il server è anche visibile come una libreria in modo da essere collegata (linkata) in un'applicazione standalone.

Data Types:

- Record di lunghezza fissa o variabile.
- Molti tipi di data type quali: i numeri con o senza segno, i char, varchar, integer, float double, text, blob, date, time, datetime, timestamp, year, set enum.

Espressioni e Funzioni:

- Gli operatori matematici sono supportati nelle query.
- Pieno supporto delle clausole GROUP BY e ORDER BY, nonché delle funzioni COUNT, COUNT DISTINCT, AVG, STD, SUM, MAX, MIN.
- Supporta il join esterno, sinistro e destro.
- Supporta gli alias su tabelle e colonne.
- DELETE, REPLACE, INSERT e UPDATE ritornano il numero delle righe che sono state modificate.
- Il comando SQL SHOW è utilizzato per ottenere informazioni su database, tabelle e indici. È in grado di “fondere” assieme tabelle differenti di database differenti in una sola query.

Sicurezza:

- Il sistema di privilegi e password è sicuro e flessibile. Le password sono sicure

perchè tutto il traffico delle password è criptato (quando ci si collega al server).

Scalabilità e limiti:

- Può manipolare grandi database. MySQL Server viene generalmente utilizzato con database che hanno 50 milioni di records. Sono concessi fino a 64 indici per tabella. Ognuno di questi indici può contenere da 1 a 16 colonne o parti di colonne.

Connettività:

- I client si possono collegare al Server SQL da ogni piattaforma utilizzando il socket TCP/IP.
- Da MySQL 4.1 in poi i server di windows supportano anche la connessione ad una memoria condivisa se lanciati con il comando `-shared-memory`.
- L'interfaccia Connector/ODBC consente a MySQL di poter essere utilizzato anche con MS Access.
- L'interfaccia Connector/J consente di supportare qualsiasi programma client java.
- MySQL Connector/NET rende possibile lo sviluppo di semplici applicazioni .Net che richiedono sicurezza, alte prestazioni.

Dislocazione:

- Il server può dare al client messaggi di errore in diverse lingue.
- Supporta una grande quantità di tipi di caratteri.
- Tutti i dati sono salvati con il set di caratteri scelto.
- L'ordinamento viene effettuato in base all'insieme dei caratteri scelto.

La stabilità di MySQL:

La radice del codice risale agli inizi degli anni 80. Questa forniva una base stabile e il formato ISAM delle tabelle utilizzato dai vecchi store engine rimase compatibile con le

versioni precedenti.

Il TcX, il predecessore del codice MySQL AB e My SQL fu utilizzato in diversi progetti fin dal 1996 senza nessun problema.

Il software per database MySQL inizialmente fu rilasciato ad un pubblico molto vasto e i nuovi utenti scoprirono velocemente parti di codice non testato, così da quel momento in poi ogni nuova release conteneva sempre meno problemi di portabilità e sempre nuovi componenti.

I problemi arrivarono quando gli utenti provarono codice della “gray zone”. Naturalmente i nuovi utenti non sapevano cosa fosse la “gray zone”, ma in qualunque caso la lista dei possibili errori contiene anche quelli di questa zona.

La struttura di MySQL Server è di tipo multi strato con moduli indipendenti. La lista qui di seguito elenca alcuni moduli e la loro valutazione:

- Tabelle InnoDB (stabili).
- Tabelle Berkeley DB (stabili).
- Ricerca Full-Text (stabile).
- MyODBC 3.51 (stabile).

La massima dimensione delle tabelle:

MySQL 3.22 aveva a disposizione 4 GB come dimensione massima delle tabelle. Con l'utilizzo di MyISAM dalla versione 3.23 la dimensione massima delle tabelle aumentò fino a 65536 terabytes. Con queste enormi dimensioni consentite per le tabelle, l'unica vera limitazione non è più dovuta a MySQL ma bensì dalle dimensioni dei file consentite dai diversi sistemi operativi. Gli store engine di tipo InnoDB, per esempio, mantengono la dimensione delle varie tabelle anche se esse possono essere create dall'unione di diversi files. Questo consente alle tabelle di superare il vincolo imposto dalla dimensione massima di un solo file. Qui di seguito è fornita una lista di sistemi operativi con il relativo limite dimensionale per i file:

- Linux 2.2-Intel 32 bit 2GB (LFS:4GB).
- Linux 2.4+ (using ext3 filesystem) 4TB.
- Solaris 9/10 16TB.
- NetWare w/NSS filesystem 8TB.
- Win32 w/ FAT/FAT32 2GB/4GB.
- Win32 w/ NTFS 2TB.
- MacOS X w/ HFS+ 2TB

Con Linux 2.2 si possono utilizzare le tabelle MyISAM più estese dei 2GB a patto di usare un file system di tipo ext2 che consente il Large File Support. Molte delle versioni di linux disponibili ora sul mercato sono basate su un kernel 2.4 o superiore il quale racchiude tutte le patch per LFS. Se si vogliono raggiungere valori ancora maggiori (dell'ordine dei petabyte) si può ricorrere all'uso di JFS e XFS. In qualunque caso la dimensione delle tabelle dipende da diversi fattori, uno di questi è sicuramente il filesystem utilizzato.

Per quanto riguarda windows è più indicato usare un file system di tipo NTFS rispetto a FAT o FAT32.

Di default MySQL crea le tabelle MyISAM con una struttura interna tale da consentire una dimensione massima di 4GB.

Se invece si ha la necessità di avere una dimensione maggiore di tali tabelle, la si può modificare usando il comando CREATE TABLE il quale supporta le opzioni AVG_ROW_LENGTH e MAX_ROWS. Oppure queste opzioni si possono anche cambiare con il comando ALTER TABLE.

Altri modi per aggirare il problema della dimensione del file:

Se la tabella di grandi dimensioni è di tipo "sola lettura" puoi usare il "myisampack" per comprimerla. Il myisampack comprime le tabelle fino al 50 per cento e può anche "unire"/"fondere" tabelle multiple in un'unica tabella.

MySQL una libreria detta libreria di MERGE che consente la portabilità di una collezione di

tabelle MyISAM che hanno un'identica struttura come un'unica tabella data appunto dal MERGE delle precedenti.

Adeguamento all'anno 2000 (il baco del millennio):

MySQLServer non ha alcun problema di adeguamento all'anno 2000, perché:

- MySQL usa le funzioni di calcolo del tempo che usa Unix. Per esempio i tipi DATE e DATETIME accettano valori fino al 9999.
- Tutte le funzioni di creazione date sono implementate in un unico file sorgente, sql/time.cc, che è stato codificato con attenzione in modo da essere resistente al baco dell'anno 2000.
- In MySQL il tipo ANNO di DATA può ospitare valori da 0 a 2155 in un solo byte. Se il campo anno fosse popolato con solo 2 cifre (ad esempio 01) SQL considera quelle 2 cifre come un indicatore di un anno compreso tra il 1970 e il 2069, quindi se noi scrivessimo 01, SQL lo interpreterebbe come il 2001

I seguenti esempi dimostrano che MySQL non ha nessun tipo di problema per la date (DATE e DATETIME) fino al 9999 e fino al 2030 per quanto riguarda il TIMESTAMP.


```
mysql> DROP TABLE IF EXISTS y2k;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE y2k (date DATE,
->                        date_time DATETIME,
->                        time_stamp TIMESTAMP);
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO y2k VALUES
-> ('1998-12-31','1998-12-31 23:59:59','1998-12-31 23:59:59'),
-> ('1999-01-01','1999-01-01 00:00:00','1999-01-01 00:00:00'),
-> ('1999-09-09','1999-09-09 23:59:59','1999-09-09 23:59:59'),
-> ('2000-01-01','2000-01-01 00:00:00','2000-01-01 00:00:00'),
-> ('2000-02-28','2000-02-28 00:00:00','2000-02-28 00:00:00'),
-> ('2000-02-29','2000-02-29 00:00:00','2000-02-29 00:00:00'),
-> ('2000-03-01','2000-03-01 00:00:00','2000-03-01 00:00:00'),
-> ('2000-12-31','2000-12-31 23:59:59','2000-12-31 23:59:59'),
-> ('2001-01-01','2001-01-01 00:00:00','2001-01-01 00:00:00'),
-> ('2004-12-31','2004-12-31 23:59:59','2004-12-31 23:59:59'),
-> ('2005-01-01','2005-01-01 00:00:00','2005-01-01 00:00:00'),
-> ('2030-01-01','2030-01-01 00:00:00','2030-01-01 00:00:00'),
-> ('2040-01-01','2040-01-01 00:00:00','2040-01-01 00:00:00'),
-> ('9999-12-31','9999-12-31 23:59:59','9999-12-31 23:59:59');
Query OK, 14 rows affected, 2 warnings (0.00 sec)
Records: 14 Duplicates: 0 Warnings: 2

mysql> SELECT * FROM y2k;
+-----+-----+-----+
| date      | date_time      | time_stamp      |
+-----+-----+-----+
| 1998-12-31 | 1998-12-31 23:59:59 | 1998-12-31 23:59:59 |
| 1999-01-01 | 1999-01-01 00:00:00 | 1999-01-01 00:00:00 |
| 1999-09-09 | 1999-09-09 23:59:59 | 1999-09-09 23:59:59 |
| 2000-01-01 | 2000-01-01 00:00:00 | 2000-01-01 00:00:00 |
| 2000-02-28 | 2000-02-28 00:00:00 | 2000-02-28 00:00:00 |
| 2000-02-29 | 2000-02-29 00:00:00 | 2000-02-29 00:00:00 |
| 2000-03-01 | 2000-03-01 00:00:00 | 2000-03-01 00:00:00 |
| 2000-12-31 | 2000-12-31 23:59:59 | 2000-12-31 23:59:59 |
| 2001-01-01 | 2001-01-01 00:00:00 | 2001-01-01 00:00:00 |
| 2004-12-31 | 2004-12-31 23:59:59 | 2004-12-31 23:59:59 |
| 2005-01-01 | 2005-01-01 00:00:00 | 2005-01-01 00:00:00 |
| 2030-01-01 | 2030-01-01 00:00:00 | 2030-01-01 00:00:00 |
| 2040-01-01 | 2040-01-01 00:00:00 | 0000-00-00 00:00:00 |
| 9999-12-31 | 9999-12-31 23:59:59 | 0000-00-00 00:00:00 |
+-----+-----+-----+
14 rows in set (0.00 sec)
```

Gli ultimi due valori della colonna TIME_STAMP sono tutti zeri perché i valori di anni impostati (2040 e 9999) sono oltre il massimo valore consentito per tale campo. Il campo TIME_STAMP varia da '1970-01-01 00:00:00' fino a '2030-01-01 00:00:00' per le macchine a 32 bit, ma arriva fino al 2106 per le macchine a 64 bit.

4 L'applicazione

4.1 Creazione della base di dati di partenza

Lo scopo del nostro database è quello di tenere traccia delle caratteristiche di un paziente, dei suoi parametri vitali e della sua posizione all'interno della struttura.

La prima fase del lavoro è stata la progettazione della base di dati che ha portato a compiere delle scelte riguardanti le tipologie di dati da conservare. Dopo un'attenta analisi siamo giunti alla conclusione che il cuore dell'applicazione doveva essere rappresentato dalla tabella Paziente, ed in particolare abbiamo evidenziato come dati critici le coordinate spaziali. Successivamente ci siamo dedicati a creare altre tabelle che potessero contenere dati di indubbia utilità, ma relativamente meno sensibili dei precedenti. Il database ha comunque subito diverse raffinazioni prima di raggiungere il livello di soddisfazione necessario.

La versione definitiva del database è mostrata di seguito:

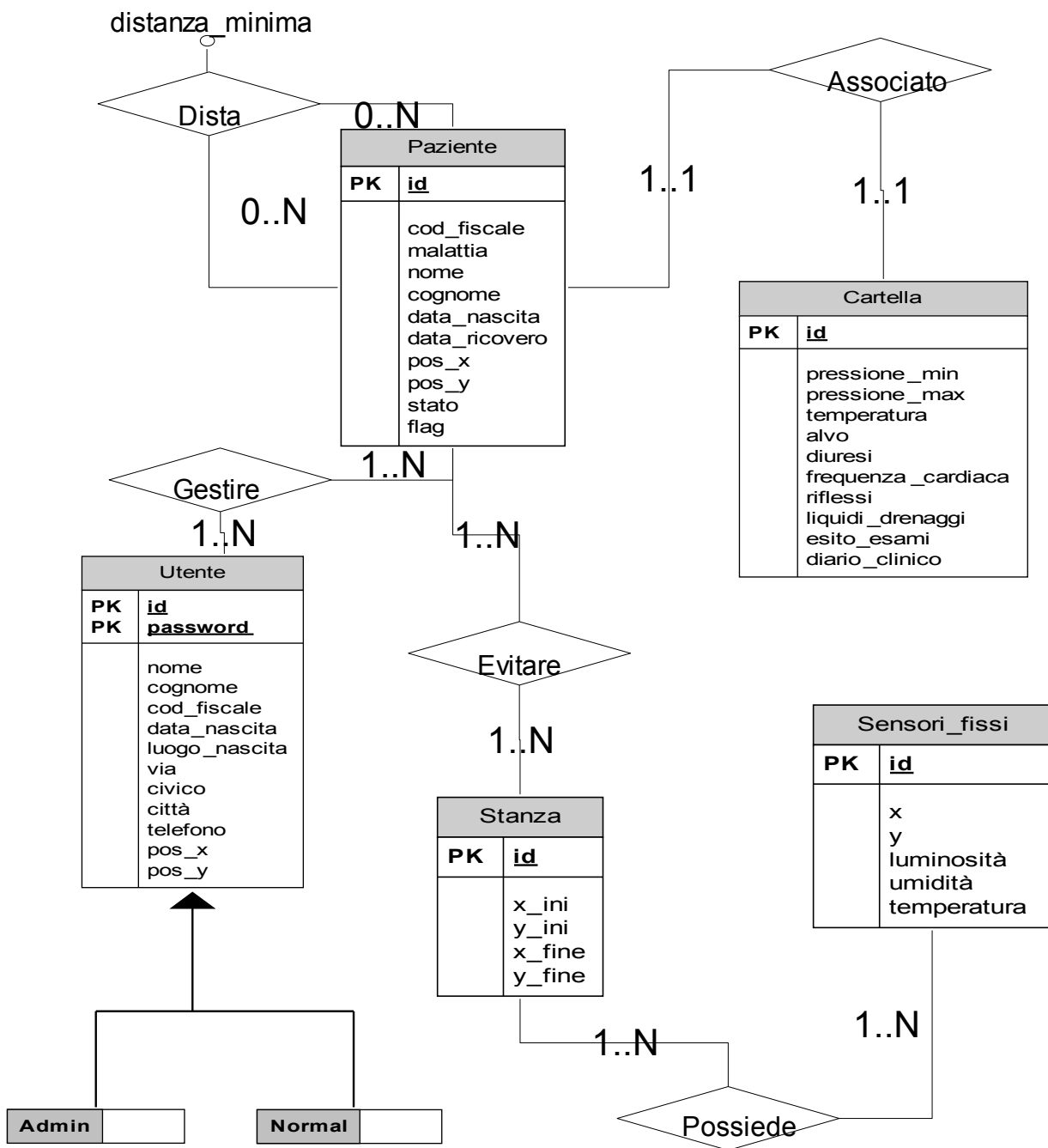


Illustrazione 4.1: Schema E-R del database

Il lavoro è proseguito con la realizzazione dello schema logico del database, la conseguente gestione ed analisi delle relazioni, le ristrutturazioni in modo da realizzare lo schema seguente:

- Paziente (id, cod_fiscale, malattia, nome, cognome, data_nascita, data_ricovero, pos_x, pos_y, stato, flag).

- Utente (id, password, nome, cognome, cod_fiscale, tipo, data_nascita, luogo_nascita, via, citta, civico, telefono, pos_x, pos_y).
- Stanza (id, x_ini, y_ini, x_fine, y_fine).
- Sensori_fissi (id, id_stanza, x, y, luminosità, umidità, temperatura).
- Cartella (id, pressione_min, pressione_max, temperatura, alvo, diuresi, frequenza_cardiaca, riflessi, liquidi_drenaggi, esito_esami, diario_clinico).
- Gestire (id_utente, password_utente, id_paziente).
- Evitare (id_paziente, id_stanza).
- Dista (id_pa, distanza_minima, id_paz).

Di seguito spiego le scelte dei campi delle tabelle e motivo alcune scelte fatte.

Considerando lo schema concettuale, l'associazione "Gestire" è stata trasformata in una tabella "ponte" tra Paziente e Utente la quale permette di capire la connessione logica che c'è tra un Utente, che può gestire più di un Paziente, e un Paziente che può essere gestito da più di un Utente (con utente si considera un'entità facente parte del personale ospedaliero).

Così come l'associazione precedente, anche "Evitare" è stata trasformata in una tabella "ponte" tra Stanza e Paziente, appunto per chiarire meglio che ad un paziente possono essere vietate tante stanze, ma anche il viceversa, cioè che una stanza può essere vietata a molti pazienti.

Nel caso di "Associato" invece abbiamo deciso di eliminare la relazione ed aggiungere, all'interno di una delle due entità associate a questa relazione la chiave primaria dell'altra; nella fattispecie la scelta è stata quella di inserire nella tabella Paziente il riferimento alla tabella Cartella.

Questa scelta è stata dettata dal fatto che l'entità Cartella è "figlia" dell'entità Paziente, in quanto senza di quest'ultima la prima non avrebbe senso di esistere.

Per quanto riguarda "Possiede" si è agito nello stesso modo visto in precedenza per "Associato".

Ultima ma non per questo meno importante la relazione “Dista” che al contrario di tutte le altre associazioni è l'unica che è dotata di un attributo.

Quest'ultima relazione rappresenta una specie di anello attorno all'entità Paziente in quanto un paziente deve poter avere vincoli di vicinanza diversi con ogni altro paziente.

4.2 Realizzazione porzione Flash

Ora che il nostro database è pronto, non resta che costruire su di esso tutta la nostra applicazione.

Uno dei tanti obiettivi di questa applicazione era quello di essere sfruttabile da ogni pc, da ogni sistema operativo, da qualunque operatore senza la necessità di un particolare tipo di training, perciò per soddisfare nel modo migliore tutti questi parametri ho pensato alla possibilità di realizzare un'applicazione basata su Web e con questa “ispirazione” mi sono messo all'opera.

Partendo dal presupposto che l'utente di questa applicazione è assolutamente digiuno di conoscenze informatiche, se non la banale navigazione internet, abbiamo puntato a realizzare l'applicazione il più user-friendly possibile. Sappiamo bene che per raggiungere tale scopo, l'unica via percorribile è quella di puntare tutto sulla renderizzazione grafica delle informazioni e degli strumenti a disposizione; così facendo ho preso in considerazione le tecniche grafiche più o meno note su cui basare l'applicazione.

La scelta non è stata affatto facile e anzi ha portato alla scelta di più di uno strumento software che potesse fare all'occorrenza.

Innanzitutto ho cercato di capire, avendo la necessità di mostrare una mappa/piantina, quale IDE potesse fare al caso mio. Dopo un'attenta analisi ho deciso di scegliere Adobe Flash (in particolare la sua versione Professional 8, ActionScript2.0), ma mi sono subito scontrato con uno dei più importanti scogli che ho trovato durante la realizzazione di questo lavoro, cioè l'assoluta assenza di conoscenze, anche di base, nell'uso di questo strumento.

Sviluppando piano piano un knowledge su Flash e su ActionScript 2.0, in particolare, ho realizzato la mappa sulla quale far comparire i pazienti nelle rispettive posizioni, e rappresentando anche lo “stato” in cui essi si trovano. Questa proprietà è stata renderizzata a video con la comparsa, in corrispondenza di ciascun paziente, di un pallino di colore verde

se lo stato risultava essere:

- MOVING
- FACE_UP
- FACE_DOWN
- STAND_UP
- FALL_RECOVERED

Da uno giallo se lo stato è:

- POSSIBLE_FALL
- CHECK_FALL

Dal colore rosso nei seguenti casi:

- FALL_ALARM
- FACE_UP, a seguito di un FALL_ALARM
- FACE_DOWN, a seguito di un FALL_ALARM

Inoltre ho anche preposto che al “click” del mouse su di un determinato paziente compare una nuvoletta contenete tutte le sue informazioni.

Per ultimo, ma non meno importante, anzi, ho realizzato sempre mediate nuvoletta, di diverso colore dalle precedenti, un controllo di intrusione non autorizzata da parte di ogni paziente nelle rispettive aree proibite.

Oltre al software che renderizza la nuvoletta e i pallini colorati, è stata fornita una versione che presenta solo una “schermata di log testuale” nel quale vengono riportati solo le righe che contengono la descrizione dei vincoli violati dai pazienti.

L'intera applicazione ha un tempo di refresh di circa 10 secondi. Tale indicazione temporale è modificabile a piacimento fino al raggiungimento di un limite fisiologico oltre il quale si entra in funzionamento critico e poco affidabile. Questo punto critico è rappresentato dal vero ed unico collo di bottiglia dell'intera applicazione, ovvero la rapidità del processore e dalla velocità con cui il programma ottiene i dati dal database.

4.3 Realizzazione porzione AJAX

Questo conclude la parte dedicata a Flash, ma non conclude di certo il mio operato, infatti si è presentato un altro scoglio da superare.

L'applicazione deve essere sempre in “primo piano” in quanto cuore dell'intero sistema di monitoraggio indoor. Si presenta allora il problema di come posso dare modo agli inservienti/medici/operatori sanitari di poter modificare ad esempio le possibili stanze da far vietare ad un paziente, le distanze che due pazienti devono rispettare tra loro senza che possano nascere problemi o situazioni pericolose e ancora la possibilità di modificare la posizione dei sensori fissi o la visualizzazione del contenuto della cartella di un paziente.

Dopo un po' di ricerche sono arrivato a conoscenza di uno “stile di programmazione”, perché a voler ben vedere è proprio questa la definizione più corretta, che veniva in contro alle mie esigenze.

La parola magica, anzi l'acronimo magico è AJAX, Asynchronous JavaScript And XML, questo stile di programmazione consente di inviare al server, in modo asincrono appunto, delle richieste e di poter poi visualizzarne le risposte all'interno della medesima pagina web senza la necessità di rinfrescarla (farne il refresh, F5).

Proprio questa peculiarità è stata la chiave per soddisfare il vincolo di “massima visibilità” della mappa pazienti.

La pagina web che contiene tutte queste componenti software, dal Flash ad ajax e non solo, è stata suddivisa in porzioni in modo tale da poter riconoscere immediatamente, da parte del programmatore, le aree soggette ad eventuali malfunzionamenti.

Per meglio chiarire al lettore verranno proposti di seguito alcuni screen-shots esplicativi di tale suddivisione.



Illustrazione 4.2: Prima parte della pagina web dell'applicazione

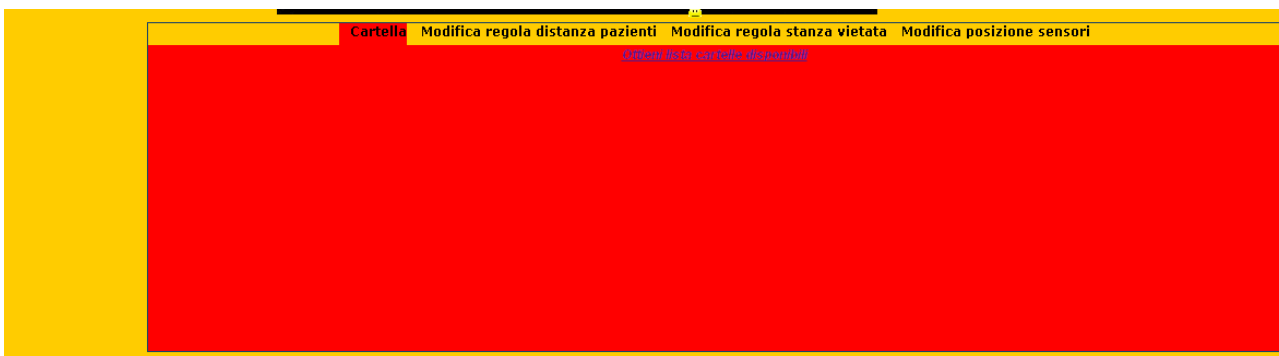


Illustrazione 4.3: Seconda parte della pagina web dell'applicazione

4.4 Realizzazione porzione java

Come complemento a questo lavoro ho realizzato un insieme di pagine jsp e servlet che hanno la funzione di facilitare l'immissione di tutti i dati (quindi il popolamento) all'interno del database di riferimento.

L'applicazione così elaborata risponde a tutti i requisiti elicitati nella fase di analisi preliminare e quindi si può concludere dicendo che tale lavoro risponde alle necessità che sono state espresse.

4.5 Simulatore

A conclusione del lavoro è stato realizzato un piccolo simulatore (anch'esso in Flash) con l'obiettivo di imitare lo spostamento di un paziente all'interno della struttura.

Più nello specifico si è creato questo simulatore con lo scopo di inserire nel database dei dati posizionali validi per un paziente e poter vedere il risultato di tali spostamenti nella nostra applicazione web.

Il simulatore, però, non si doveva limitare ad inserire dati validi nel database, ma quei dati dovevano avere un certo “significato”.

La specifica per la realizzazione di questo componente era relativamente chiara, doveva ottenere una quantità a caso (random) di punti (coordinate X e Y) che dovevano appartenere ad una curva.

Tale simulatore è stato realizzato immaginando che la curva da cui ricavare i punti non fosse una semplice retta, ma una forma d'onda più complessa, riconducibile ad una sinusoidale.

Questo simulatore è stato creato pensando allo spostamento random di un movieclip, da un punto sorgente, preso a caso, ad un punto destinazione, anch'esso preso a caso. La traiettoria con cui si muoverà questo movieclip è completamente random e una volta che il clip raggiunge il punto di destinazione, l'applicazione ritorna alle condizioni iniziali; generando altri punti a caso come partenza ed arrivo della traiettoria.

La struttura del codice è molto semplice, in quanto si è deciso di inserire tutti i “comportamenti” all'interno di un unico frame.

L'idea per trovare la traiettoria random è quella di costruire un polinomio di terzo grado pensando di usare l'asse “X” pari alla retta che congiunge i punti iniziale e finale, mentre le radici del polinomio sono il punto iniziale, il punto finale ed una serie di punti intermedi calcolati di volta in volta.

Una volta calcolati questi punti bisogna adattarli alla rappresentazione in flash usando la seguente regola:

$$x = x' \cdot \cos \alpha - y' \cdot \sin \alpha$$

$$y = x' \cdot \sin \alpha + y' \cdot \cos \alpha,$$

dove x ed y sono le normali coordinate e x' ed y' sono le coordinate ruotate di un angolo pari all'angolo che c'è tra punto di origine e destinazione della simulazione vedi figura:

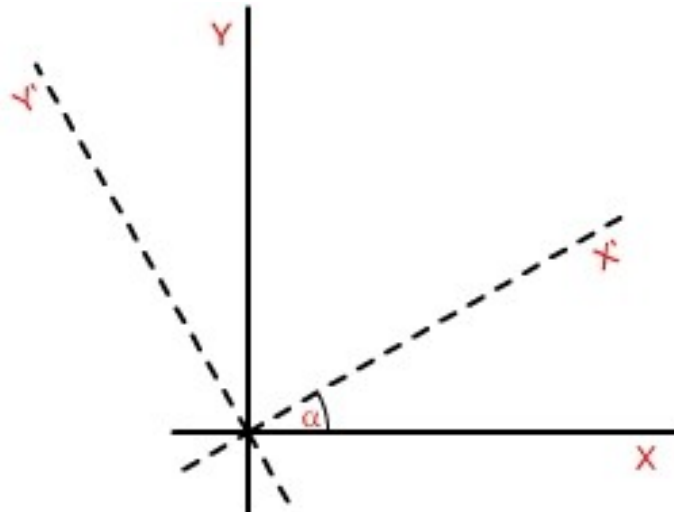


Illustrazione 4.4: Angolo tra punto iniziale e finale

di seguito vediamo lo schema dell'andamento dei punti della simulazione (il punto X_{sRFA} è il nostro punto di destinazione):

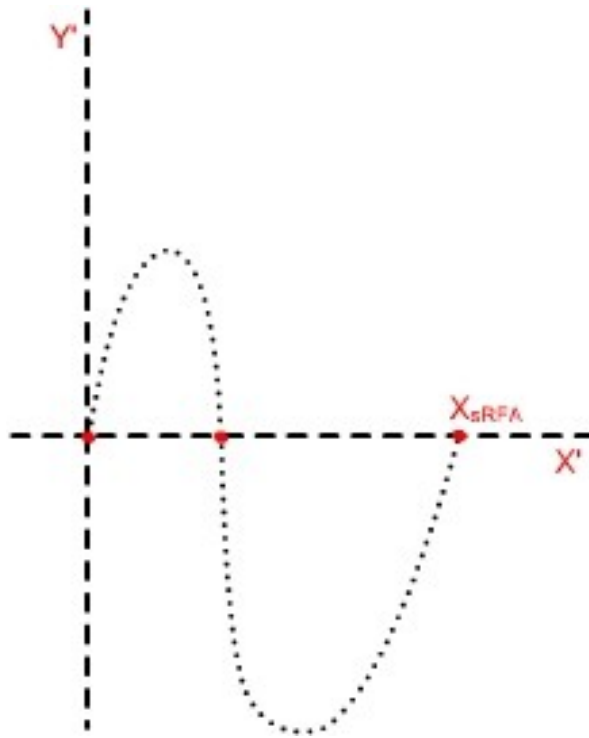


Illustrazione 4.5: Andamento simulazione

5 Use Case

5.1 Pagina principale per utenti Normal



Illustrazione 5.1:Pagina principale per utenti Normal

Elementi di rilievo presenti nella figura sono:

1. Pulsante di redirectione al LOGOUT e comando per far apparire o scomparire la porzione AJAX.
2. Porzione di applicazione che rappresenta il cuore dell'applicazione in quanto è la mappa di visualizzazione pazienti; nella fattispecie possiamo notare come i pazienti visualizzati possiedono uno stato “sano” in quanto gli smile sono verdi e sorridenti, ed inoltre possiamo notare come uno di essi violi il vincolo di restrizioni de stanze ad esso associato.
3. Sezione AJAX la quale contiene diverse “schede” con diverse operazioni implementate; tale sezione sfruttando ajax non necessita di refresh della pagina per la visualizzazione delle operazioni e dei relativi risultati.

5.2 Pagina principale per utenti Normal versione "Line"



Illustrazione 5.2: Pagina principale utenti Normal versione "Line"

Elementi di rilievo presenti nella figura sono:

1. Pulsante di redirezione al LOGOUT e comando per far apparire o scomparire la porzione AJAX.
2. Porzione di applicazione che rappresenta il cuore dell'applicazione in quanto è il "log testuale" che propone a video le sole informazioni riguardanti i pazienti che violano le regole di intrusione e vicinanza, rispettivamente con stanze e altri pazienti.
3. Sezione AJAX la quale contiene diverse "schede" con diverse operazioni implementate; tale sezione sfruttando ajax non necessita di refresh della pagina per la visualizzazione delle operazioni e dei relativi risultati.

5.3 Pagina principale per utenti Admin



Illustrazione 5.3: Pagina principale utenti Admin

Elementi di rilievo presenti nella figura sono:

1. Pulsanti di redirectione alle altre pagine web che consentono l'immissione facilitata di dati nel database (inserisci cartella, inserisci paziente, inserisci sensore, inserisci stanza, inserisci utente ed inserisce nuove regole di intrusione stanza) oltre che al LOGOUT e comando per far apparire o scomparire la porzione AJAX.
2. Porzione di applicazione che rappresenta il cuore dell'applicazione in quanto è la mappa di visualizzazione pazienti; nella fattispecie possiamo notare come i pazienti visualizzati possiedono uno stato "sano" in quanto gli smile sono verdi e sorridenti, ed inoltre possiamo notare come uno di essi violi il vincolo di restrizioni di stanze ad esso associato.
3. Sezione AJAX la quale contiene diverse "schede" con diverse operazioni implementate; tale sezione sfruttando ajax non necessita di refresh della pagina per la visualizzazione delle operazioni e dei relativi risultati.

5.4 Pagina principale per utenti Admin versione "Line"



Illustrazione 5.4: Pagina principale utenti Admin versione "Line"

Elementi di rilievo presenti nella figura sono:

1. Pulsanti di redirectione alle altre pagine web che consentono l'immissione facilitata di dati nel database (inserisci cartella, inserisci paziente, inserisci sensore, inserisci stanza, inserisci utente ed inserisce nuove regole di intrusione stanza) oltre che al LOGOUT e comando per far apparire o scomparire la porzione AJAX.
2. Sezione AJAX la quale contiene diverse "schede" con diverse operazioni implementate; tale sezione sfruttando ajax non necessita di refresh della pagina per la visualizzazione delle operazioni e dei relativi risultati.
3. Porzione di applicazione che rappresenta il cuore dell'applicazione in quanto è il "log testuale" che propone a video le sole informazioni riguardanti i pazienti che violano le regole di intrusione e vicinanza, rispettivamente con stanze e altri pazienti.

5.5 Pagine inserimento dati cartella

The image shows a yellow background with a form for entering patient data. The form consists of 11 white input fields, each with a label in blue text to its right. The labels are: 'id (numerico)', 'pressione max', 'pressione min', 'temperatura', 'alvo (1=SI : 0=NO)', 'diuresi', 'frequenza cardiaca', 'riflessi', 'quantità di liquidi drenati', 'esito esami', and 'diario clinico'. Below the input fields are three buttons: 'Invia', 'Reset', and a link 'Torna alla Home'. Three blue numbers are overlaid on the image: '1' is positioned to the right of the first four input fields; '2' is positioned to the right of the 'Invia' button; '3' is positioned to the right of the 'Torna alla Home' link.

Illustrazione 5.5: Pagina dell'applicazione per l'inserimento facilitato dei parametri della cartella paziente

Elementi di rilievo presenti nella figura sono:

1. Form di inserimento dei vari dati che caratterizzano la cartella del paziente (id, pressione massima, pressione minima, temperatura, alvo, diuresi, frequenza cardiaca, riflessi, quantità liquidi drenati, esito esami, diario clinico).
2. Pulsanti per la sottomissione (Invia) o per la cancellazione (Reset) dei dati inseriti nella form.
3. Pulsante per la redirectione alla pagine di visualizzazione mappa.

5.6 Pagina inserimento dati paziente

The image shows a yellow background with a form for entering patient data. The form consists of 13 input fields, each with a label in blue text. The labels are: 'id (numerico)', 'codice fiscale', 'malattia', 'nome', 'cognome', 'data nascita (AAAA-MM-GG)', 'data ricovero (AAAA-MM-GG)', 'coordinata X della posizione del paziente (0 per inserire posizione di default)', 'coordinata Y della posizione del paziente (0 per inserire posizione di default)', 'inserire flag (valore 0/1; 0 per il valore di default)', and 'id cartella associata al paziente (numerico)'. Below the input fields are two buttons: 'Invia' and 'Reset'. At the bottom left is a link 'Torna alla Home'. A large blue number '1' is placed to the right of the first seven input fields. A large blue number '2' is placed to the right of the 'Invia' and 'Reset' buttons. A large blue number '3' is placed to the right of the 'Torna alla Home' link.

Illustrazione 5.6: Pagina dell'applicazione per l'inserimento facilitato dei parametri del paziente

Elementi di rilievo presenti nella figura sono:

1. Form di inserimento dei vari dati che caratterizzano la cartella del paziente (id, codice fiscale, malattia, nome, cognome, data di nascita, data di ricovero, coordinata X del paziente, coordinata Y del paziente, flag, id cartella associata al paziente).
2. Pulsanti per la sottomissione (Invia) o per la cancellazione (Reset) dei dati inseriti nella form.
3. Pulsante per la redirectione alla pagine di visualizzazione mappa.

5.7 Pagine inserimento dati sensori fissi

id (numerico)

id stanza di riferimento (numerico)

coordinata X del sensore

coordinata Y del sensore

luminosità

umidità

temperatura

Invia

Reset

[Torna alla Home](#)

Illustrazione 5.7: Pagina dell'applicazione per l'inserimento facilitato dei parametri dei sensori fissi

Elementi di rilievo presenti nella figura sono:

1. Form di inserimento dei vari dati che caratterizzano il sensore(id, id stanza di riferimento, coordinata X del sensore, coordinata Y del sensore, luminosità, umidità, temperatura).
2. Pulsanti per la sottomissione (Invia) o per la cancellazione (Reset) dei dati inseriti nella form.
3. Pulsante per la redirectione alla pagine di visualizzazione mappa.

5.8 Pagina di inserimento nuovo utente

The image shows a user registration form on a yellow background. The form contains the following fields and controls:

- id (numerica)
- password (non deve iniziare con un numero)
- Scegli il tipo (dropdown menu)
- nome
- cognome
- codice fiscale
- data nascita (AAAA-MM-GG)
- luogo di nascita
- via
- civico
- città
- telefono
- coordinata X dell'utente
- coordinata Y dell'utente
-
-
- [Torna alla Home](#)

Three blue numbers are overlaid on the form: '1' is next to the 'nome' field, '2' is next to the 'Invia' button, and '3' is next to the 'Torna alla Home' link.

Illustrazione 5.8: Pagina dell'applicazione per l'inserimento facilitato dei parametri dell'utente

Elementi di rilievo presenti nella figura sono:

1. Form di inserimento dei vari dati che caratterizzano il nuovo utente (id, password, tipo (Admin o Normal), nome, cognome, data di nascita, luogo di nascita, via, civico, città, telefono, coordinata X dell'utente, coordinata Y dell'utente).
2. Pulsanti per la sottomissione (Invia) o per la cancellazione (Reset) dei dati inseriti nella form.
3. Pulsante per la redirectione alla pagine di visualizzazione mappa.

5.9 Pagina per la definizione dei confini di stanza



Illustrazione 5.9: Pagina dell'applicazione per l'inserimento facilitato delle dimensioni della stanza

Elementi di rilievo presenti nella figura sono:

1. Form di inserimento dell'identificativo della nuova stanza (id).
2. Pulsanti per la sottomissione (Invia) dei dati inseriti nella form e dei dati presi dalla selezione compiuta sull'immagine.
3. Mappa sulla quali si può tracciare mediato mouse un quadrilatero i cui bordi rappresentano le dimensioni della stanza e le relative coordinate.

5.10 Pagina per la definizione della regole di intrusione in una stanza



Illustrazione 5.10: Pagina dell'applicazione per l'inserimento facilitato delle regole di violazione stanza

Elementi di rilievo presenti nella figura sono:

1. Form dinamica per inserimento dell'identificativo del paziente che deve evitare una stanza e della relativa stanza (id_paziente, id_stanza).
2. Pulsanti per la sottomissione (Conferma) dei dati inseriti nella form e per il ritorno alla pagina dell'applicazione di monitoring.

6 Evaluation e Performance

Questo applicativo essendo basato su Web ha come unico e vero vincolo prestazionale, la necessità di supportare il non troppo leggero componente Flash.

Una delle controindicazioni di questo componente è il fatto che è molto esigente come risorse, sia di processore che di memoria ram; per il suo corretto funzionamento.

Come base di partenza e comune per ogni genere di valutazione dell'applicativo è la necessità di avere un browser web, configurato in modo tale da eseguire piccoli script lato client (presenti nella porzione AJAX) e avere un Flash Player aggiornato.

Di seguito vengono proposti un paio di simulazioni effettuate su due macchine differenti, ciò è stato fatto per mostrare l'andamento del carico dell'applicazione sul sistema.

Come parametri di testing sono stati presi:

- prima simulazione “base” con circa 5 pazienti
- seconda simulazione “intermedia” con circa 30 pazienti
- terza simulazione “over” con circa 50 pazienti

I benchmark di seguito proposti sono stati presi considerando che tutti i controlli che l'applicazione deve effettuare, cioè:

- controllo intrusione stanza
- controllo distanza pazienti
- controllo stato pazienti

vengono tutti effettuati ogni 100ms.

6.1 Simulazione "base"

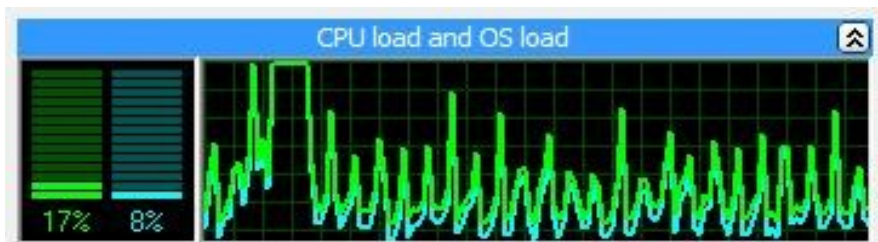


Illustrazione 6.1: Andamento dell'applicazione con pochi pazienti

Questa immagine rappresenta l'andamento del carico di cpu e sistema operativo quando è in funzione l'applicazione.

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Windows Vista Home Basic
- CPU: Sempron Mobile 3400+
- Frequenza di elaborazione: 1,8GHz
- Architettura 32 bit
- Core fisici: 1
- Core virtuali: 1
- Ram 2x512MB
- Scheda Grafica: Nvidia
- Memoria Grafica: 64MB condivisa

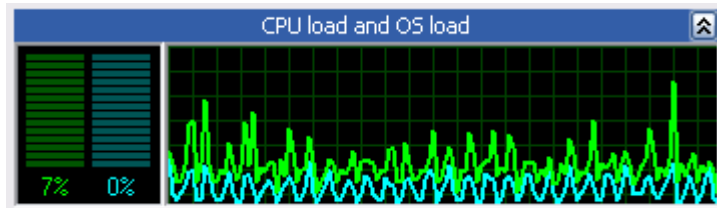


Illustrazione 6.2: Andamento dell'applicazione con pochi pazienti

Queste immagini rappresentano l'andamento del carico di cpu quando è in funzione l'applicazione (doppio grafico in quanto la CPU è dual-core).

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Windows XP sp3
- CPU: Intel Centrino Core Duo T2050
- Frequenza di elaborazione:1,6GHz
- Architettura 32 bit
- Core fisici: 2
- Core virtuali: 2
- Ram 2x512MB
- Scheda Grafica: Intel GMA 950
- Memoria Grafica: 64MB condivisa

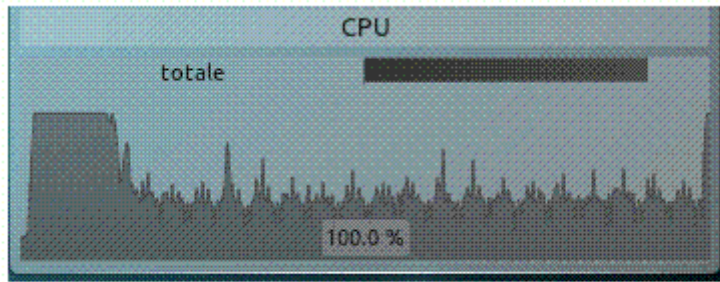


Illustrazione 6.3: Andamento dell'applicazione con pochi pazienti

Questa immagine rappresenta l'andamento del carico di cpu con sistema operativo Linux quando è in funzione l'applicazione.

I picchi iniziali e finali al 100% sono dovuti alla partenza dell'applicazione e al momento della cattura dell'immagine, il corpo centrale ha una media di carico di circa in 37%.

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Linux - Kubuntu 11.04
- CPU: Sempron Mobile 3400+
- Frequenza di elaborazione:1,8GHz
- Architettura 32 bit
- Core fisici: 1
- Core virtuali: 1
- Ram 2x512MB
- Scheda Grafica: Nvidia
- Memoria Grafica: 64MB condivisa

6.2 Simulazione "intermedia"

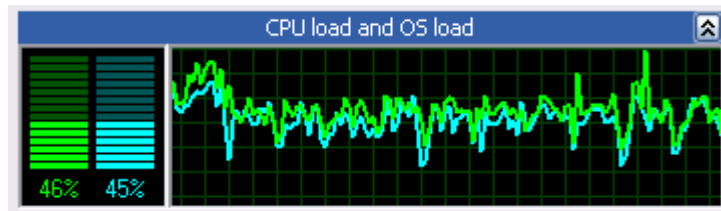


Illustrazione 6.4: Andamento applicazione caso intermedio

Queste immagini rappresentano l'andamento del carico di cpu quando è in funzione l'applicazione (doppio grafico in quanto la CPU è dual-core).

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Windows XP sp3
- CPU: Intel Centrino Core Duo T2050
- Frequenza di elaborazione: 1,6GHz
- Architettura 32 bit
- Core fisici: 2
- Core virtuali: 2
- Ram 2x512MB
- Scheda Grafica: Intel GMA 950
- Memoria Grafica: 64MB condivisa

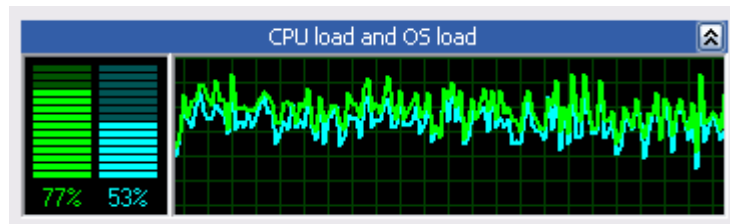


Illustrazione 6.5: Andamento applicazione caso intermedio

Questa immagine rappresenta l'andamento del carico di cpu e sistema operativo quando è in funzione l'applicazione.

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Windows Vista Home Basic
- CPU: Sempron Mobile 3400+
- Frequenza di elaborazione: 1,8GHz
- Architettura 32 bit
- Core fisici: 1
- Core virtuali: 1
- Ram 2x512MB
- Scheda Grafica: Nvidia
- Memoria Grafica: 64MB condivisa

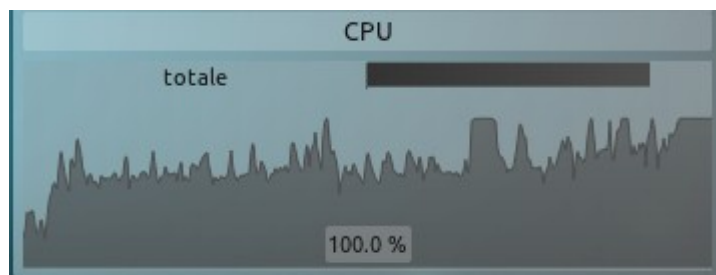


Illustrazione 6.6: Andamento applicazione caso intermedio

Questa immagine rappresenta l'andamento del carico di cpu con sistema operativo Linux quando è in funzione l'applicazione.

I picchi iniziali e finali al 100% sono dovuti alla partenza dell'applicazione e al momento della cattura dell'immagine, il corpo centrale ha una media di carico di circa in 70%.

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Linux - Kubuntu 11.04
- CPU: Sempron Mobile 3400+
- Frequenza di elaborazione:1,8GHz
- Architettura 32 bit
- Core fisici: 1
- Core virtuali: 1
- Ram 2x512MB
- Scheda Grafica: Nvidia
- Memoria Grafica: 64MB condivisa

6.3 Simulazione "over"

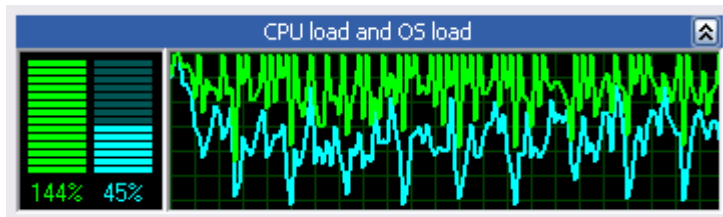


Illustrazione 6.7: Andamento applicazione caso over

Queste immagini rappresentano l'andamento del carico di cpu quando è in funzione l'applicazione (doppio grafico in quanto la CPU è dual-core).

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Windows XP sp3
- CPU: Intel Centrino Core Duo T2050
- Frequenza di elaborazione: 1,6GHz
- Architettura 32 bit
- Core fisici: 2
- Core virtuali: 2
- Ram 2x512MB
- Scheda Grafica: Intel GMA 950
- Memoria Grafica: 64MB condivisa

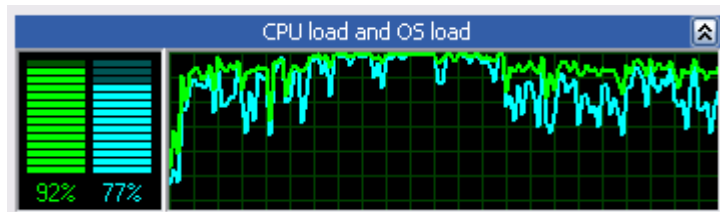


Illustrazione 6.8: Andamento applicazione caso over

Questa immagine rappresenta l'andamento del carico di cpu e sistema operativo quando è in funzione l'applicazione.

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Windows Vista Home Basic
- CPU: Sempron Mobile 3400+
- Frequenza di elaborazione: 1,8GHz
- Architettura 32 bit
- Core fisici: 1
- Core virtuali: 1
- Ram 2x512MB
- Scheda Grafica: Nvidia
- Memoria Grafica: 64MB condivisa

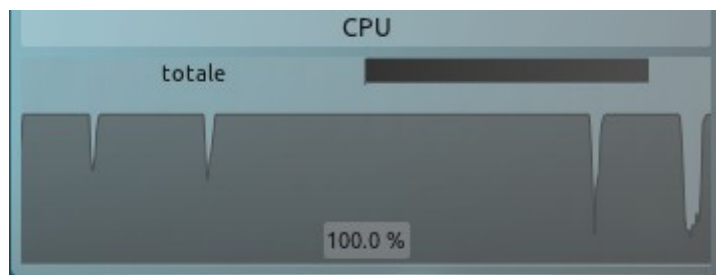


Illustrazione 6.9: Andamento applicazione caso over

Questa immagine rappresenta l'andamento del carico di cpu con sistema operativo Linux quando è in funzione l'applicazione.

In questa simulazione linux ha fatto vedere tutti i suoi limiti nell'uso di Flash, infatti la cpu lavora sempre al massimo.

Flash in linux parte sempre penalizzato in quando deve un po' essere “simulato” e questo sappiamo che porta a circa un “raddoppio” della necessità delle risorse cpu e ram in primis.

Le caratteristiche hardware e software del computer su cui è stato testato sono:

- Sistema operativo: Linux - Kubuntu 11.04
- CPU: Sempron Mobile 3400+
- Frequenza di elaborazione:1,8GHz
- Architettura 32 bit
- Core fisici: 1
- Core virtuali: 1
- Ram 2x512MB
- Scheda Grafica: Nvidia
- Memoria Grafica: 64MB condivisa

Un'analisi dei dati ottenuti dalle simulazioni pare doverosa e porta a fare una riflessione; l'applicazione risulta essere funzionante in ogni condizione, ma sapendo che Flash Player esige molte risorse, ed avendo testato su macchine di fascia bassa, possiamo ritenere soddisfacente il risultato derivato dalle varie curve di carico e possiamo ritenerci soddisfatti per il lavoro svolto.

7 Conclusioni e Sviluppi futuri

7.1 Conclusioni

Valutando il lavoro svolto nel suo complesso posso asserire con certezza che tutti i vincoli sono stati rispettati, per scrupolo ricordo che i maggiori vincoli erano:

- Ottenere massima visibilità della mappa pazienti.
- Rendere disponibili funzionalità di modifica parametri senza per questo perdere di vista la mappa.
- Rendere accessibile a tutti tale applicativo, senza necessità di installare un particolare software e tanto meno avere la necessità di effettuare del training agli utenti.

Non solo siamo riusciti a fare in modo che questi vincoli fossero soddisfatti, ma abbiamo anche sfruttato il mondo del web per offrire un'applicazione platform-independent, cioè abbiamo un'applicazione che funziona su qualunque sistema operativo, su qualunque struttura hardware, sia essa PC o MAC; in conclusione può essere usata da qualunque dispositivo, anche da dispositivi mobili quali Smart-phone e PDA; in pratica abbiamo raggiunto la massima fruibilità possibile.

7.2 Sviluppi futuri

Il nostro applicativo raggiunge un buon grado di usabilità e raggiunge il più ampio grado di portabilità, tuttavia si può sempre migliorare.

Una delle principali migliorie apportate risulta essere quella riguardante la tecnologia usata per il rendering della mappa, cioè Flash; sapendo che la caratteristica del player più famoso del web è quella di essere esigente a livello hardware, si può pensare di rimodellare tale porzione di codice con una in AJAX, ispirandosi a GOOGLE e alle sue famose GOOGLE-MAPS.

Ovviamente, un sistema come LAURA nel suo complesso, potrebbe essere rimaneggiato ed adattato anche a contesti estremamente differenti rispetto a quello clinico. Alcuni semplici esempi potrebbero riguardare il settore penitenziario, quello veterinario, quello sportivo, quello domotico...

Le aspettative riguardanti questo progetto sono estremamente ampie!

8 Bibliografia

- [1] <http://laura.como.polimi.it/Home.html>.
- [2] M. Cesana L. P. Borsani P. Tarrío F. Salice A. Redondi, M. Tagliasacchi. Laura - localization and ubiquitous monitoring of patients for healthcare support. IEEE International Workshop on Advances in Positioning and Location-Enabled Communications (APLEC), 09 2010.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002.
- [4] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal. Gps: location-tracking technology. *Computer*, 35(4):92–94, March 2002.
- [5] P. Bonato. Wearable sensors/systems and their impact on biomedical engineering. *IEEE Engineering in Medicine and Biology Magazine*, 22(3):18–20, Maggio/Giugno 2003.
- [6] R. Guy C. Haase. *Filthy Rich Clients - Developing Animated and Graphical Effects for Desktop Java Applications*. Prentice Hall, 2007.
- [7] R. Cutler and L. S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):781–796, August 2000.
- [8] F. Fleuret, F. J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008.
- [9] M. Tagliasacchi L. P. Borsani A. Redondi J. R. Casar P. Tarrío, M. Cesana. An energy-efficient strategy for combined rss-pdr indoor localization. 7th IEEE PerCom Workshop on Mobile Pervasive Wireless Networking, 2011.
- [10] V. Stanford. Pervasive computing goes the last hundred feet with rfid systems. *IEEE Pervasive Computing*, 2(2):9–14, April-June 2003.
- [11] Bruce W. Perry. *Ajax trucchi e segreti*. O'Reilly, HOPS tecniche nuove, settembre 2006.
- [12] Jen deHaan, Peter de Haan. *ActionScript 2.0 per Macromedia Flash8 documentazione ufficiale*. Mondadori Informatica. Febbraio 2006.

9 Ringraziamenti

Questa tesi rappresenta il culmine di un impegno che è iniziato anni fa e che non saremmo certamente stati in grado di portare a termine da soli; ci sentiamo quindi di ringraziare alcune persone che ci sono state vicine durante questo percorso.

Il primo pensiero va sicuramente alle nostre famiglie, senza il cui sostegno economico e soprattutto morale non avremmo neanche potuto intraprendere questa esperienza.

Ci sono state anche altre persone fondamentali durante gli anni di impegno accademico, cioè gli amici e compagni di corso (in ordine alfabetico): Ing. Raimondo Riccardo, Ing. Rosa Marco, Ing. Salvia Marco, Ing. Scalvinoni Alberto, Spagnuolo Pio, Dott. Tosto Giuseppe e tanti altri i quali oltre a condividere con me una risata, spesso per stemperare la tensione pre-esame, hanno anche rappresentato un mio punto di riferimento a livello didattico.

Ovviamente meritano un ringraziamento anche l'ing. Cesana Matteo, l'ing. Redondi Alessandro e l'ing. Borsani Luca che ci hanno fornito le capacità, i materiali e le conoscenze necessarie per lo sviluppo di questa tesi.

10 Appendice A

10.1 Diagrammi UML

Pagine “Client”; pagine HTML:

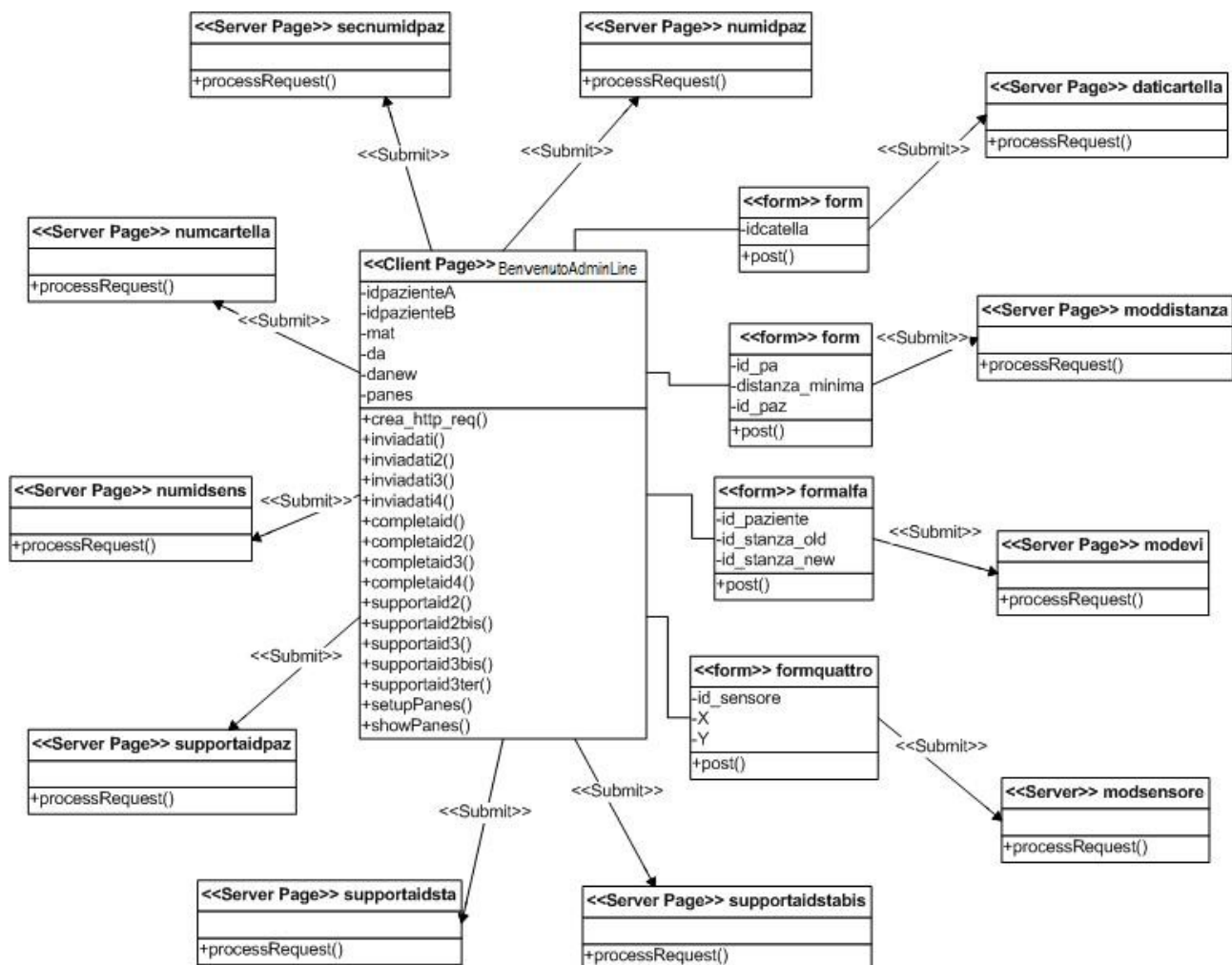


Illustrazione 10.1: Pagina BenvenutoAdminLine

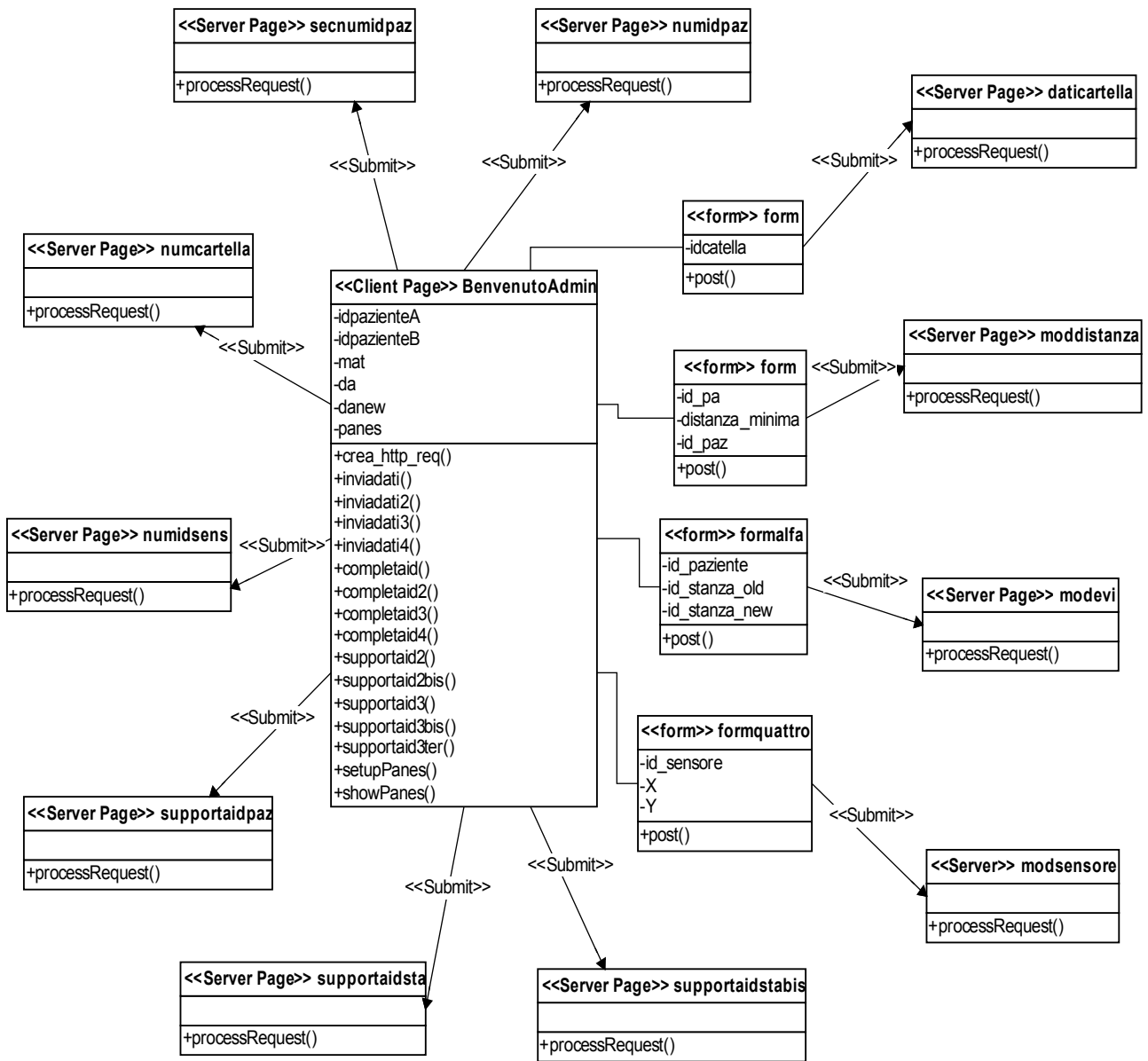


Illustrazione 10.2: Pagina BenvenutoAdmin

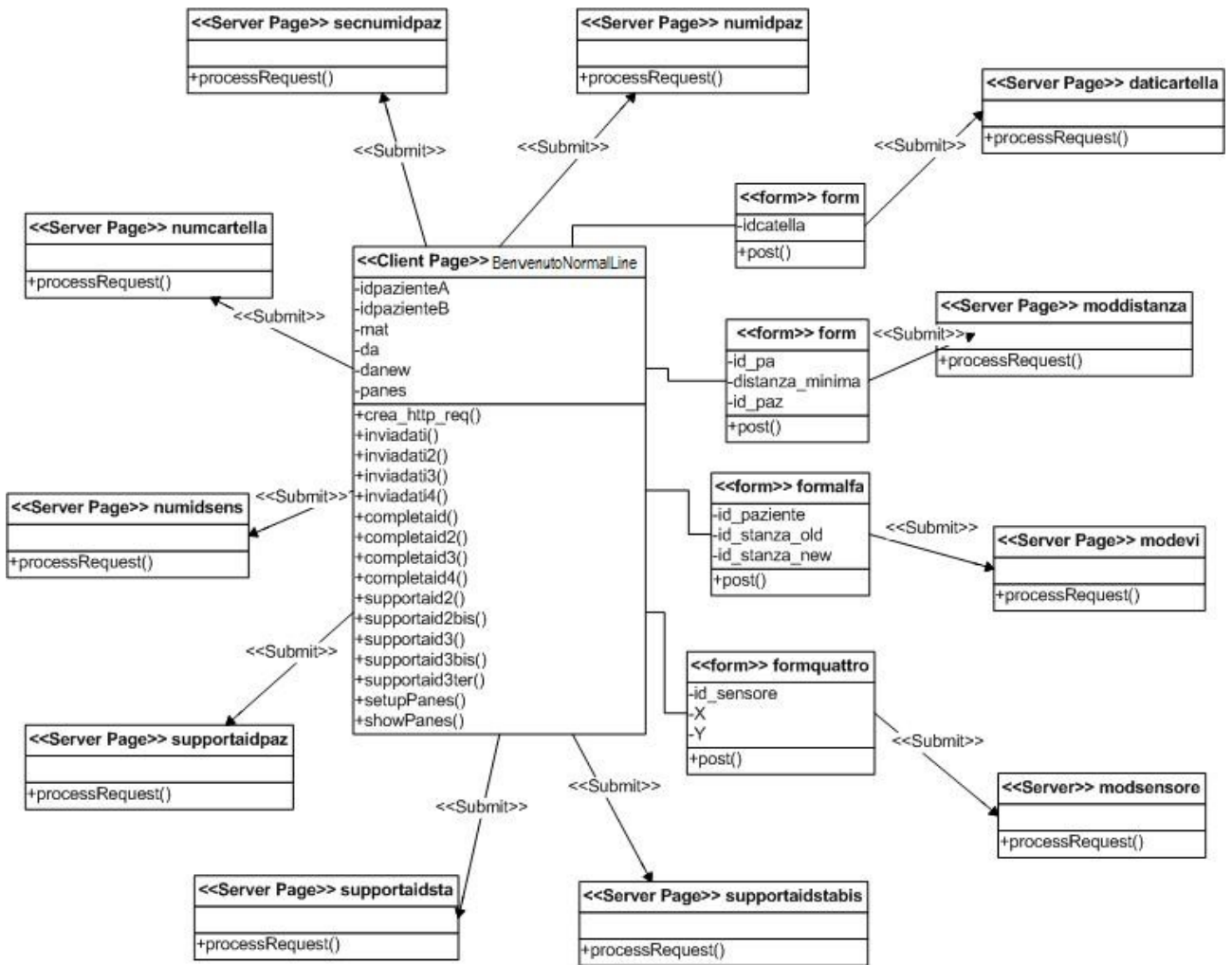


Illustrazione 10.3: Pagina BenvenutoNormalLine

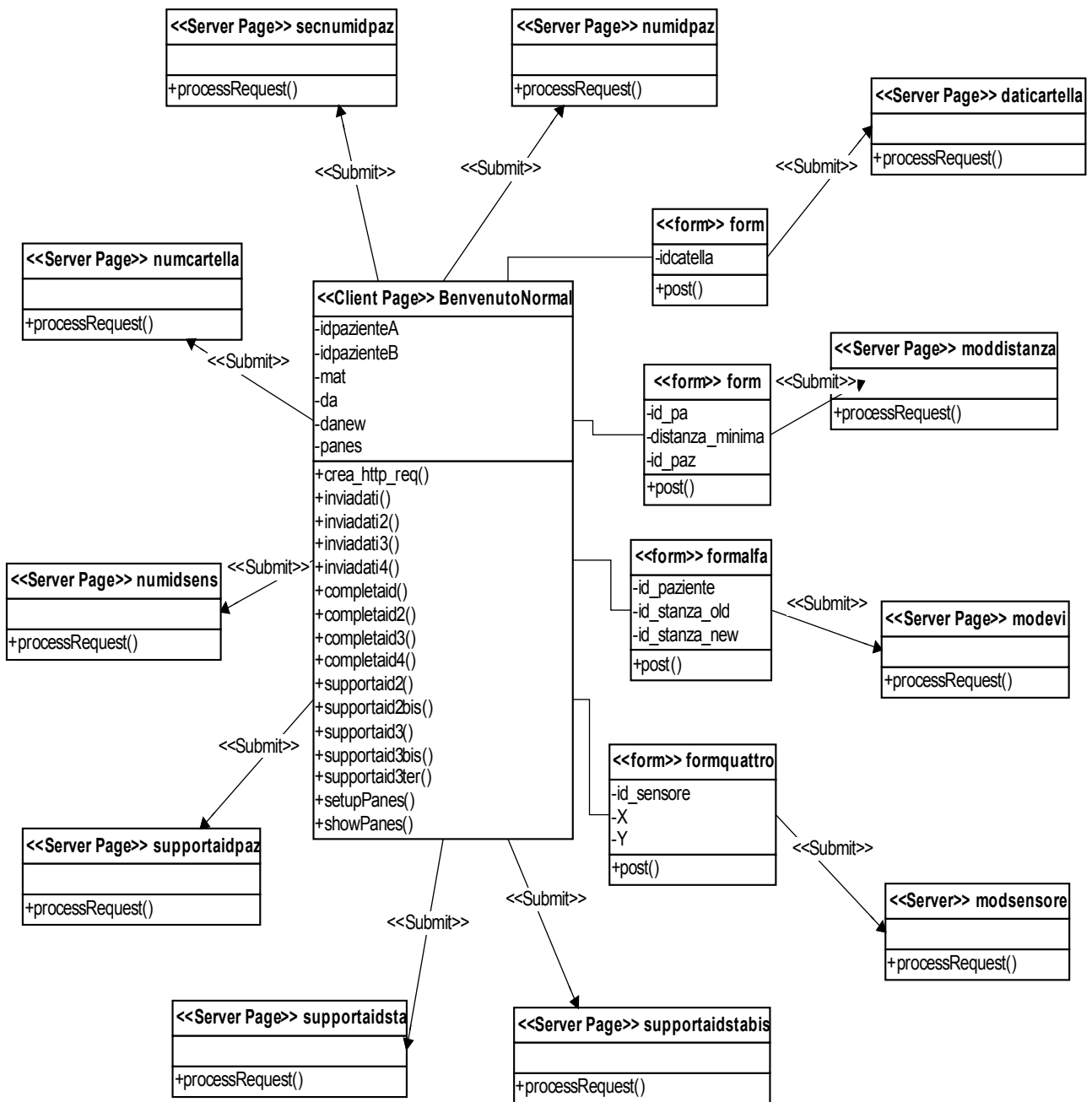


Illustrazione 10.4: Pagina BenvenutoNormal

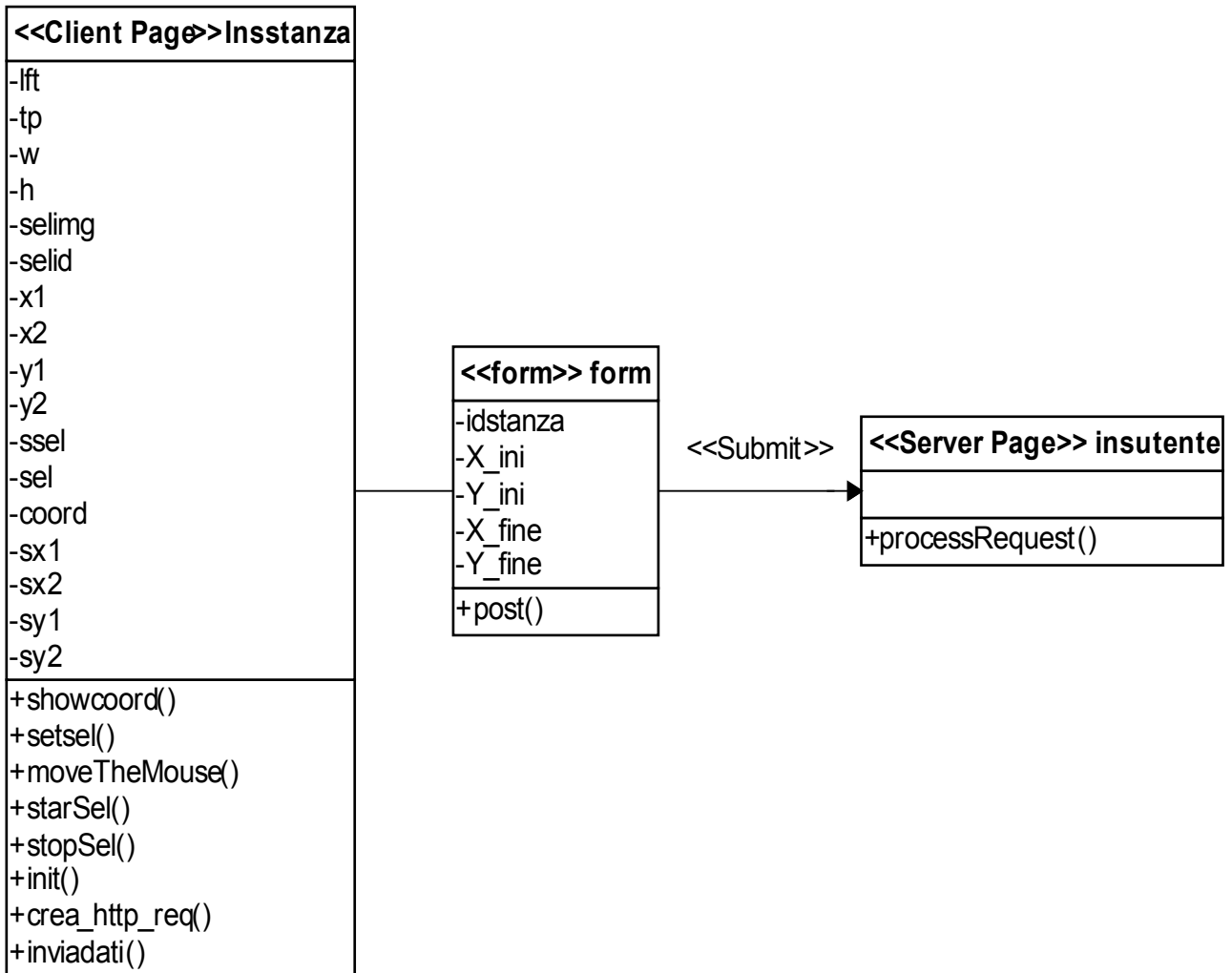


Illustrazione 10.5: Pagina Inserimento Stanza

Pagine “server”; cioè pagine JSP/Servlet:

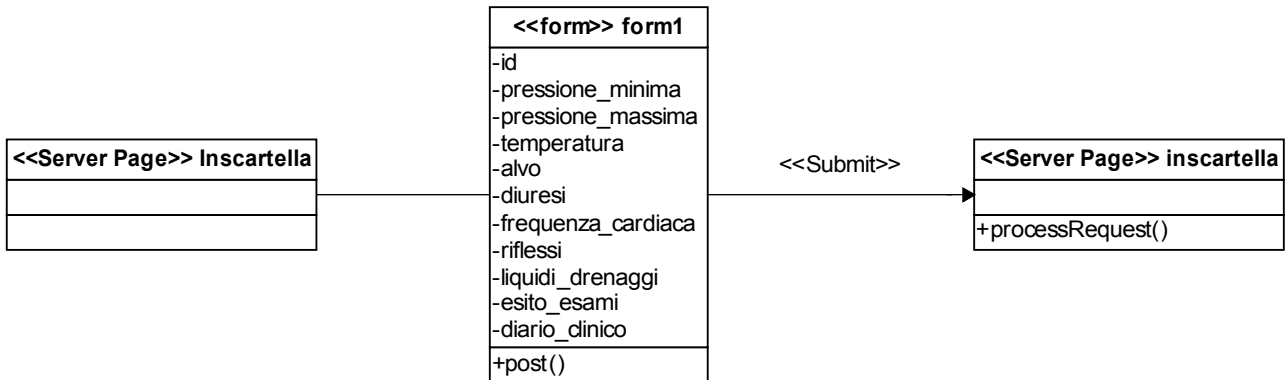


Illustrazione 10.6: Pagina Inserimento Cartella clinica del Paziente

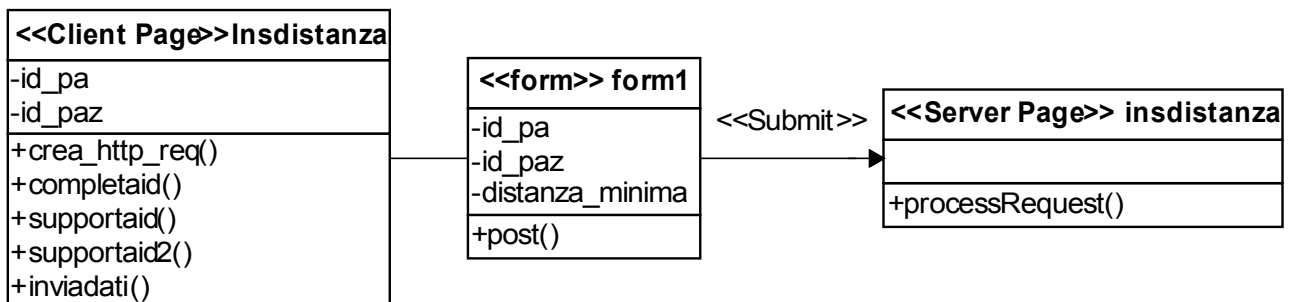


Illustrazione 10.7: Pagina Inserimento vincolo di distanza tra pazienti

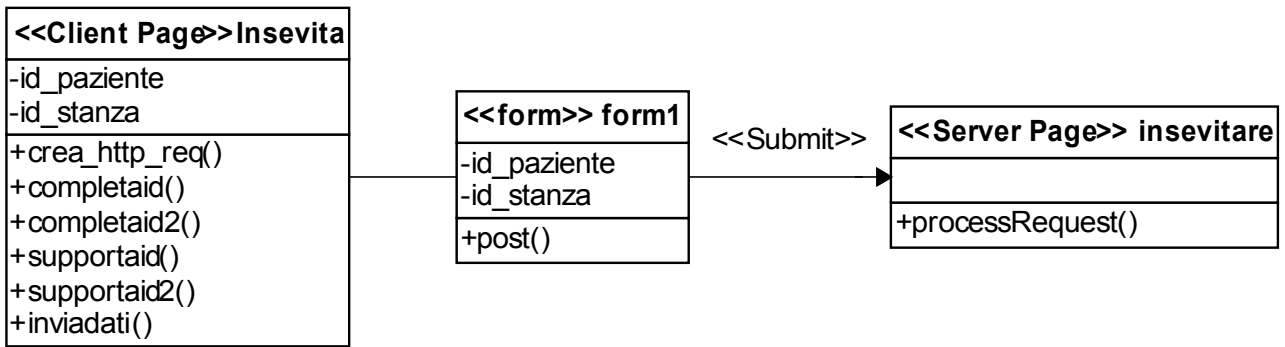


Illustrazione 10.8: Pagina di Inserimento Stanza da Evitare

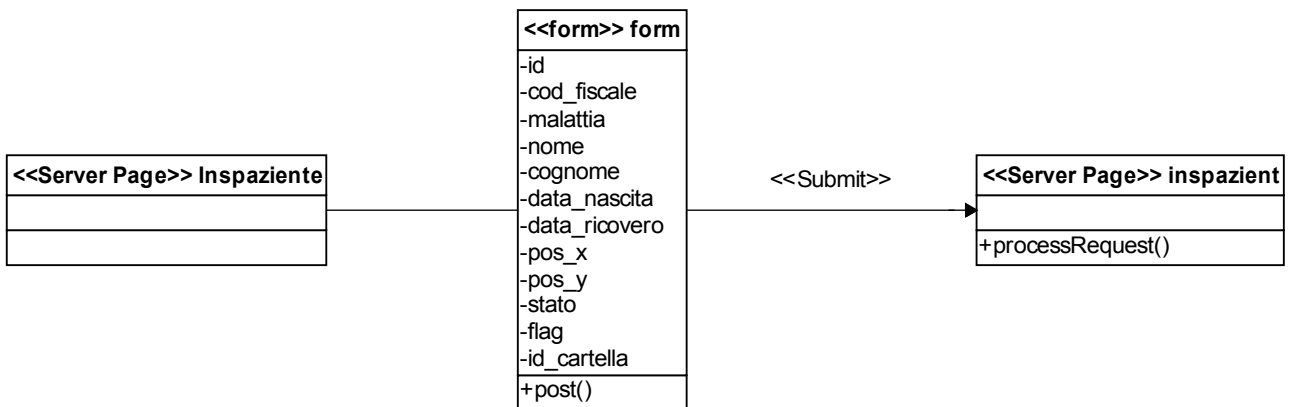


Illustrazione 10.9: Pagina Inserimento dati Paziente

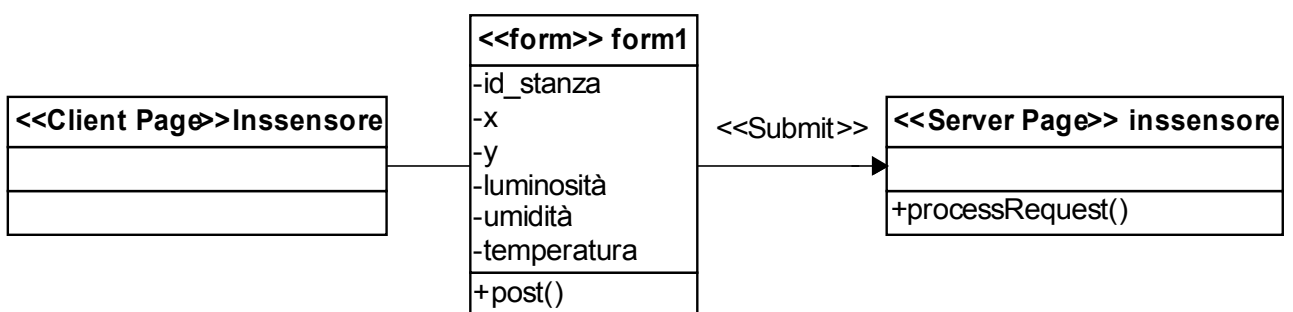


Illustrazione 10.10: Pagina Inserimento dati Sensore

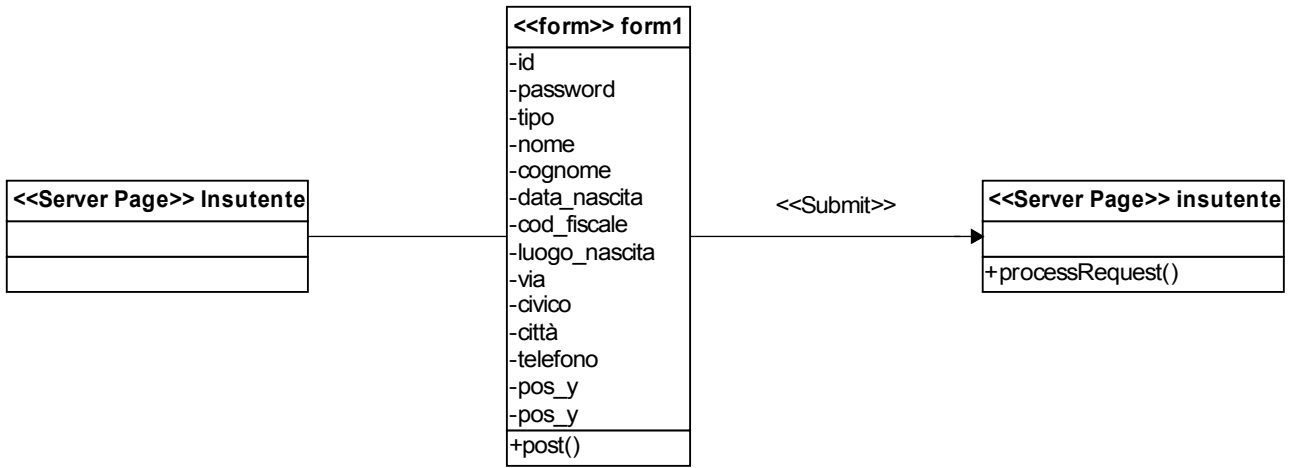


Illustrazione 10.11: Pagina Inserimento dati Utente