

# UNIVERSITA' DEGLI STUDI DI PADOVA

Facoltà di Scienze  
Matematiche Fisiche e Naturali

Corso di Laurea in Informatica

Tesi di Laurea

Sistema di Esportazione Dati Alimenti  
per il Ministero della Salute

Relatore: *Ch. mo Prof. Alessandro Sperduti*

Laureando: *Piero Dalle Pezze*

Anno Accademico 2005-2006



## Indice :

Introduzione	5
1. L'Agenzia ARPAV	7
1.1. Il Servizio SIRAV	8
2. Strumenti di lavoro	10
3. Presentazione generale del prodotto	12
3.1. Descrizione del progetto	12
3.2. L'interfaccia grafica	14
3.3. Semplicità d'utilizzo per l'utente	18
3.4. Descrizione di un record esportato	18
4. Analisi, progettazione e realizzazione del software	21
4.1. Il modello di ciclo di vita	21
4.2. La pianificazione delle attività	22
4.3. Gli standard di progetto adottati	26
4.3.1. Standard di progettazione architetturale	26
4.3.2. Standard di documentazione del codice	26
4.3.3. Standard di denominazione di entità e relazioni	27
4.3.4. Standard di programmazione	27
4.3.5. Standard di lavoro	28
4.4. Ambientamento	28
4.5. Analisi dei requisiti	29
4.6. Progettazione architetturale	32
4.7. Progettazione di dettaglio	39
4.8. Realizzazione	47
4.9. Verifica	48
5. Gestione della qualità	50

Conclusioni	52
Appendice	54
Piano di progetto	55
Piano di qualifica	70
Analisi dei requisiti	92
Specifica tecnica	114
Definizione di prodotto	129
Manuale utente	149
Ringraziamenti	162
Glossario	163
Bibliografia	172

## ***Introduzione***

Questa tesi rappresenta una relazione della mia attività di stage, presso l'Agenzia ARPAV. Questa necessitava di un sistema software che potesse automatizzare il processo di esportazione dei dati, riguardanti i campioni e le analisi degli alimenti, in una codifica stabilita dal Ministero della Salute Italiana. In particolare, era necessario che il prodotto provvedesse alla generazione di un file, contenente le informazioni anagrafiche relative ad ogni campione alimentare autorizzato ed i risultati delle analisi effettuate su questi. In questo modo, il Ministero della Salute, una volta ricevuto il file di esportazione dall'Agenzia, sarebbe in grado di importare immediatamente questi dati nel proprio sistema, senza ricorrere ad alcuna ulteriore decodifica.

L'attività di stage che ho svolto, comprendeva l'analisi, la progettazione e la realizzazione di tale sistema ed eventualmente l'inserimento nel database ARPAV di alcune tabelle ausiliarie per la decodifica. Avendo avuto la possibilità da parte del tutor aziendale Paolo Zambotto, di poter lavorare anche a casa, sono riuscito a completare lo stage in otto settimane. In questo periodo, ho potuto, in primo luogo, consolidare le conoscenze apprese nel corso di Ingegneria del Software, sia dal punto di vista gestionale, che da quello produttivo, realizzando un'adeguata documentazione, in secondo luogo, apprendere il linguaggio di programmazione PHP, linguaggio richiesto dall'Agenzia per l'implementazione del sistema.

La tesi è strutturata per gradi di complessità, in modo tale da poter distinguere chiaramente *cosa* fa il software da me prodotto e *come* esso funziona.

I capitoli 1 e 2 presentano una descrizione succinta dell'ambiente presso il quale ho lavorato e quali sono stati gli strumenti che ho avuto a disposizione per lo svolgimento di questo progetto.

Il capitolo 3 propone il software dal punto di vista dell'utente, illustrandone lo scopo e le funzionalità. Esso, quindi, fornisce inizialmente una descrizione della

genesì e dell'obiettivo del progetto, introducendo in seguito il lettore, alle caratteristiche salienti del prodotto.

Il capitolo 4 delinea le modalità di sviluppo del progetto, secondo i principi dell'ingegneria del software, mantenendo un certo grado di astrazione. Dopo una presentazione degli aspetti puramente gestionali, quali il modello di ciclo di vita adottato, la pianificazione ed il controllo delle attività, si espongono gli aspetti produttivi, sottolineando gli standard applicati, le problematiche coinvolte, l'analisi dei requisiti, la progettazione architetturale e di dettaglio, la realizzazione e la verifica.

Il capitolo 5 pone l'attenzione alla qualità del software ed alla misurazione di questa. Esso illustra quindi la procedura impiegata per la gestione della qualità ed il tentativo di stabilire una scala di misura che consentisse di quantificare il grado qualitativo del mio progetto.

Infine, per un ulteriore approfondimento delle specifiche del software, sono forniti in appendice i documenti formali esterni prodotti in ciascuna fase dello stage, ossia il piano di progetto, il piano di qualifica, l'analisi dei requisiti, la specifica tecnica, la definizione di prodotto ed il manuale utente. Il glossario è stato integrato con quello di questa tesi.

## ***Capitolo 1. L'Agenzia ARPAV***

L'acronimo ARPAV significa **A**genzia **R**egionale per la **P**revenzione e **P**rotezione Ambientale del Veneto. L'Agenzia, istituita con la [Legge Regionale n°32 del 18 ottobre 1996](#) ed operativamente attiva a partire dal 3 ottobre 1997, è sorta con due scopi strettamente connessi:

- la protezione: attraverso i controlli ambientali che tutelano la salute della popolazione e la sicurezza del territorio;
- la prevenzione: attraverso la ricerca, la formazione, l'informazione e l'educazione ambientale.

L'ARPAV è un'agenzia regionale dotata di autonomia amministrativa, organizzativa, tecnica e contabile. Opera sulla base di piani triennali e di un programma annuale.

In essa sono presenti diverse figure professionali che garantiscono un approccio multidisciplinare ai compiti dell'Agenzia medesima, scambiandosi informazioni e innovazioni.

Come si può notare in figura 1, l'Agenzia dispone di una struttura modulare ampia che si occupa di diversi ambiti. Una Direzione Centrale governa le tre aree funzionali:

- [area amministrativa](#), ossia la gestione del personale, del bilancio e del patrimonio, nonché ogni altra attività amministrativa di carattere unitario;
- [area tecnico-scientifica](#), ossia la promozione, la programmazione e la produzione di servizi connessi alle attività tecnico-scientifiche dell'ARPAV;
- [area ricerca e informazione](#), ossia la promozione e lo sviluppo della ricerca e

dell'avanzamento delle tecnologie per la tutela dell'ambiente, la gestione delle attività di informazione, formazione, sensibilizzazione ed educazione ambientale e la gestione del sistema informativo regionale ambientale.

L'area Tecnico-Scientifica/Ricerca e Informazione definisce i rapporti con i sette dipartimenti provinciali della regione Veneto, con gli staff dell'Agenzia e con i servizi di cui dispone.

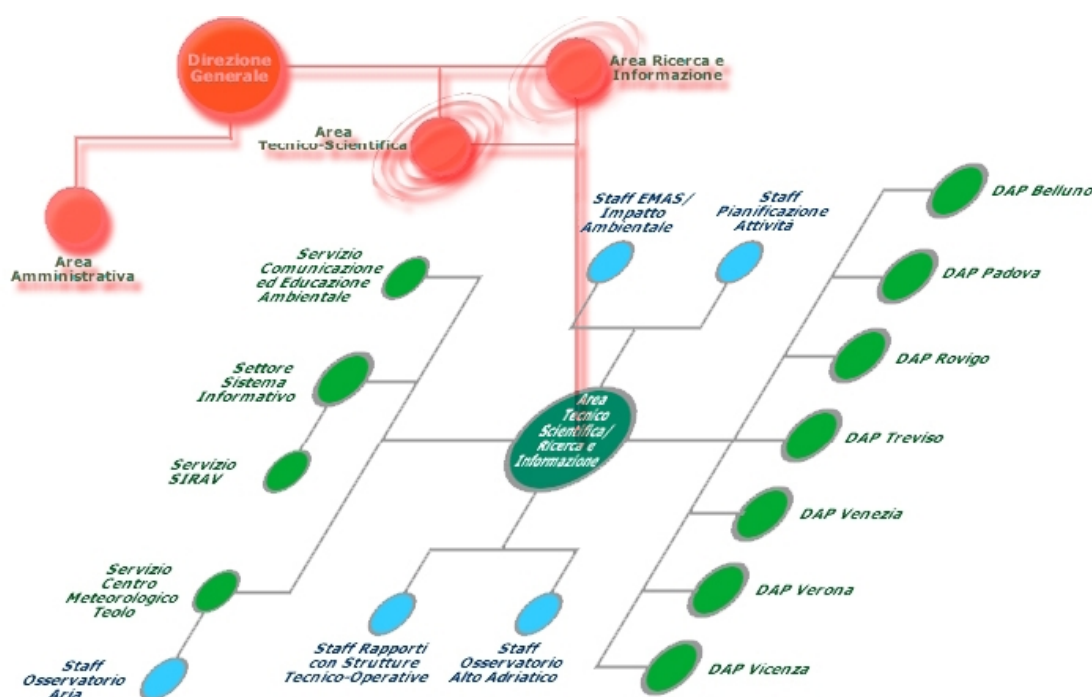


Figura 1. Struttura dell'Agenzia ARPAV<sup>1</sup>.

### 1.1. Il Servizio SIRAV

L'attività di stage si è svolta presso l'ufficio SIRAV, che è parte integrante del Settore Sistema Informativo. Il servizio SIRAV si occupa di:

- organizzazione, coordinamento e responsabilità sulle attività, sulle strutture e sulle risorse umane e strumentali afferenti al Servizio;
- organizzazione dello sviluppo dei contenuti del Sistema Informativo Ambientale e delle regole per il suo popolamento, in collaborazione con gli

1. Questa figura è stata concessa dall'Agenzia ARPAV.



Osservatori, i Dipartimenti Provinciali e i soggetti esterni (Province, Regioni...);

- standardizzazione, sviluppo e manutenzione dei sistemi informatici;
- amministrazione delle risorse informatiche dell'Agenzia (macchine, reti), supporto e formazione all'uso di sistemi e procedure informatiche;
- organizzazione della raccolta delle informazioni ambientali e dello sviluppo dei contenuti e degli applicativi per la produzione di report integrati sullo stato dell'ambiente a livello regionale;
- organizzazione e gestione del Punto Focale Regionale (PFR) di raccolta delle informazioni territoriali e ambientali nel Veneto per il SINA (Sistema Informativo Nazionale Ambiente);
- organizzazione, definizione di standard e distribuzione dell'informazione cartografica territoriale e ambientale.

## ***Capitolo 2. Strumenti di lavoro***

L'Agenzia ARPAV ha messo a disposizione, per lo svolgimento dell'attività di stage, una postazione di lavoro costituita da un Personal Computer Intel Pentium IV 1700 MHz, 130 MB di RAM con installato il sistema operativo Microsoft Windows 2000 Professional.

Oltre al Personal Computer precedentemente citato, ho utilizzato anche il mio notebook personale Intel Pentium IV 1700 MHz, 500 MB di RAM con installato il sistema operativo Microsoft Windows XP Home Edition.

Tra i linguaggi di programmazione è stato deciso di usare i seguenti: PHP 5.1.2, CSS, HTML, PL/SQL ed SQL. PHP è un linguaggio di programmazione molto popolare tra i progetti Open Source del web. Esso agisce sul lato server ed è principalmente utilizzato per la realizzazione di pagine web dinamiche e l'interrogazione di un database. A partire dalla versione 5, il linguaggio PHP supporta pienamente la programmazione ad oggetti. Ciò lo rende quindi molto versatile e facilmente utilizzabile da chi conosce già questo paradigma. Per interpretare correttamente questo linguaggio, l'Agenzia ARPAV utilizza il web server Apache 2.0.49. CSS è un linguaggio utilizzato per definire il foglio di stile, ossia un file contenente le descrizioni e le caratteristiche dei formati che si intendono utilizzare in un insieme di pagine web. HTML è un linguaggio per descrivere come la pagina web deve essere visualizzata. PL/SQL è un linguaggio che integra con la programmazione procedurale, il linguaggio usato per l'interrogazione di database, SQL.

L'adozione del linguaggio PHP, voluto dall'Agenzia, è stata rilevante in quanto, questo strumento possiede molte funzionalità per l'accesso a database, per le date, per la manipolazione delle stringhe e dispone di una sintassi simile al C. Esso quindi comprendeva tutti i mezzi necessari per la realizzazione del mio progetto,

senza dover ricorrere alla definizione di nuove strutture dati o funzioni aggiuntive. Grazie ai corsi di Programmazione ed alla documentazione ufficiale presente in rete sul sito <http://www.php.net/>, ho potuto apprendere le principali caratteristiche di questo linguaggio facilmente ed in poco tempo, senza ritardi significativi nello stato di avanzamento del progetto.

Per quanto concerne la realizzare dei diagrammi UML, si è utilizzato l'ambiente di sviluppo integrato (IDE) EclipseUML 3.1 Free Edition e il programma Inkscape 0.43, sotto licenza GNU, per l'elaborazione.

L'ARPAV utilizza il database Oracle 8.1.6 per la memorizzazione dei dati e gli applicativi QueryBuilder, SQLPlus Worksheet, TOAD 8.5 per il reperibilità, la modifica, la cancellazione e l'inserimento dei dati. Durante questo stage, ho quindi avuto la possibilità di operare su un sistema molto complesso a cui non sono mai stato fino ad allora abituato e di utilizzare della strumentazione sofisticata.

Per la realizzazione del codice sorgente, ho utilizzato gli editor di testo: GNU Emacs 21.3.1 e XEmacs 21.4.19, entrambi sotto licenza GNU. Ho deciso di adoperare tali editor, seppure considerati primitivi da taluni, per la tangibile versatilità e rapidità d'uso mediante la sola tastiera.

Per quanto riguarda la stesura dei documenti, ho utilizzato l'editor OpenOffice 2.0, anziché il pacchetto MS Office utilizzato dall'ARPAV, per ragioni di maggior stabilità ed efficienza.

I documenti in modalità sola lettura sono nel formato PDF (Adobe Reader 7.0.8).

Infine, per l'esecuzione dei test e la visualizzazione del prodotto finale, è stato utilizzato il browser per la navigazione su Internet Mozilla Firefox 1.5.0.4. Sono inoltre stati utilizzati anche i diffusi browser Microsoft Internet Explorer 6.0 e Opera 9, per testare la corretta visualizzazione del prodotto software.

## ***Capitolo 3. Presentazione generale del prodotto***

Questo capitolo intende introdurre il lettore alla comprensione dell'origine e della funzione di questo prodotto software. Si presenterà quindi il concepimento del problema dal quale questo progetto è sorto ed il funzionamento dal punto di vista dell'utente. Nel capitolo successivo, invece si condurrà il lettore ai dettagli dell'analisi, della progettazione e della realizzazione di questo software.

### ***3.1. Descrizione del progetto***

Con lo scopo di importare i campioni e le analisi alimentari delle singole Agenzie Regionali in maniera uniforme, il Ministero della Salute Italiana ha fornito, ad ogni Agenzia<sup>2</sup>, un software che permette di generare un file di esportazione in una specifica codifica. Ogni Agenzia Regionale ha dunque il compito di far pervenire il proprio file di esportazione al Ministero della Salute il quale, quindi, deve soltanto occuparsi dell'importazione di questi file ricevuti nel proprio sistema, senza ricorrere ad alcuna decodifica.

Il personale incaricato di utilizzare il programma ministeriale precedentemente citato, ha il compito di individuare i dati richiesti nel proprio sistema e di inserirli manualmente nel programma.

Nell'attività di stage, mi è stato proposto come progetto, un sistema software che potesse automatizzare questo processo. L'acronimo SEDAMS da me ideato, denota questo **Sistema di Esportazione Dati Alimenti per il Ministero della Salute**. Nel corso di questa tesi utilizzerò sempre la sigla.

Il progetto SEDAMS consentirà all'Agenzia ARPAV, di esportare i dati riguardanti le analisi effettuate su campioni di alimenti, presenti nei database provinciali nello specifico formato descritto nel documento SA-99-D1006-015 del Ministero della Salute.

---

2. Per la regione Veneto, tale Agenzia è l'ARPAV.

I benefici più evidenti con l'utilizzo di questo prodotto sono:

1. L'efficienza del processo di esportazione. Non è più necessario incaricare una persona a tale scopo.
2. Il conseguente risparmio economico.
3. La garanzia di correttezza dei dati esportati.
4. La stabilità del processo.

Il flusso logico delle informazioni è il seguente:

1. Nel luogo di origine, le stazioni di monitoraggio, prelevano i dati.
2. I prelievi passano attraverso un livello di aggregazione ed integrazione provinciale.
3. I prelievi, quindi, arrivano a costituire la Banca Dati Ambientale, integrata di livello regionale.
4. Dalle banche dati regionali, il flusso informativo viene trasmesso agli organismi nazionali ed al pubblico.

Il progetto SEDAMS, si occuperà, operando dalla postazione base regionale, di prelevare le informazioni dai sette database provinciali dell'Agenzia ARPAV. In particolare, quindi, opera nel terzo e nel quarto stadio del flusso logico delle informazioni, interfacciandosi con il database Oracle principale dell'ARPAV e con i database Oracle provinciali della regione Veneto.

Il prodotto consente al responsabile per l'esportazione, di ottenere un file, contenente l'insieme dei campioni<sup>3</sup> e delle analisi effettuate per ognuno di questi<sup>4</sup>, codificato come previsto nel documento ministeriale SA-99-D1006-015. Il software SEDAMS può essere reperibile in rete soltanto da personale autorizzato, ossia da coloro che possiedono l'accesso diretto ai database dell'Agenzia e che si sono già

---

3. I campioni devono essere di tipo alimentare.

4. I campioni e le analisi da esportare devono essere stati approvati.

connessi al server centrale.

Il software SEDAMS provvede, inoltre, alla generazione del file dei campioni che non sono stati esportati a causa di dati mancanti nel database stesso, in modo tale da poterli individuare e correggere all'interno del database dal personale ARPAV.

### **3.2. L'interfaccia grafica**

Il software è composto da due finestre principali, la prima per l'inserimento dati, la seconda per visualizzare l'evoluzione dell'esportazione.

Di seguito, sono presentate nel dettaglio i contenuti di queste due finestre.

1. **Finestra di inserimento dati.** Per effettuare un'esportazione, l'utente deve compilare tutti i seguenti campi:
  - Data Inizio. Rappresenta la data a partire dalla quale si intende effettuare l'esportazione. Occorre specificare il giorno, il mese e l'anno.
  - Data Fine. Rappresenta la data entro la quale si intende effettuare l'esportazione. Occorre specificare il giorno, il mese e l'anno con le stesse modalità definite nel campo Data Inizio<sup>5</sup>.
  - Login. E' la login dell'utente che intende effettuare l'esportazione. Tale login è definita internamente alla rete ARPAV.
  - Password. E' la password dell'utente che intende effettuare l'esportazione. Anch'essa è definita internamente alla rete ARPAV.
  - Seleziona il server. Occorre selezionare il server dal quale si intende effettuare l'esportazione<sup>6</sup>.
  - Avvia Esportazione. E' il bottone che permette di avviare l'esportazione.
  
2. **Finestra dell'evoluzione dell'esportazione.** Dopo aver premuto il bottone "Avvia Esportazione", il software presenta in modo completo i dettagli della sua esecuzione. Di seguito sono presentate le informazioni che il

---

5. I campi Data Inizio e Data Fine definiscono il periodo di esportazione. Entrambe le date devono essere valide e la prima deve precedere la seconda.

6. I nomi dei server sono così composti: Database (DB) + Sigla della Provincia del Veneto + (LIMS).

software SEDAMS rende disponibili.

- Titolo. Ricorda all'utente quale server è stato selezionato.
- Periodo di esportazione. Ricorda all'utente il periodo entro il quale il software sta effettuando l'esportazione.
- Inizializzazione del path. Specifica le azioni del software nel filesystem. In particolare, il software deve testare l'esistenza delle cartelle *export/* e *not\_export/*. All'interno di queste due cartelle, devono essere create due ulteriori cartelle con il nome YYMMDD (ossia anno, mese e giorno) correnti (es. 060515) che definiscono la data di esportazione. In questo modo, il software evita la sovrascrittura dei files se sono stati creati in due date distinte, dal momento che apparterranno a due cartelle con nome distinto. Il software, invece, sovrascriverà i files nel caso in cui si effettuasse più di un'esportazione nella stessa data e dallo stesso server. Se occorresse effettuare più di un'esportazione dallo stesso server nello stesso giorno, l'utente dovrà manualmente copiare il file prima di effettuare la successiva esportazione.
- Stato della transazione. Definisce se il software è riuscito a stabilire una connessione al server. In seguito provvede a stampare, per ogni campione prelevato, l'esito della sua esportazione e quante analisi possiede. Se un campione non viene esportato, a causa di nessuna analisi selezionata, significa che il campo AV\_CAMPO1 della tabella COMPONENT è nullo per tutte le analisi di quel campione. Infine, dopo aver esportato tutti i campioni e le loro analisi per il periodo specificato, il software dichiara la chiusura della connessione.
- Visualizzazione delle statistiche dell'esportazione. Dopo aver terminato l'esportazione, il software SEDAMS presenta le statistiche principali. In particolare vengono riportate le seguenti informazioni:
  - ♦ Numero dei campioni che sono stati prelevati.
  - ♦ Numero dei campioni che sono stati esportati.
  - ♦ Numero dei campioni che sono stati prelevati ma non esportati.

- ◆ Numero delle analisi che sono state prelevate ed esportate.
  - ◆ Numero medio di analisi per campione (prelevato).
  - ◆ Numero medio di analisi per campione (esportato).
  - ◆ Dimensione del file di esportazione (in KB).
  - ◆ Dimensione del file dei campioni non esportati (in KB).
  - ◆ Durata della transazione (in ore, minuti, secondi).
  - ◆ Path assoluto del file di esportazione.
  - ◆ Path assoluto del file dei campioni non esportati.
- Files di esportazione. Dopo aver eseguito un'esportazione, il software SEDAMS presenta l'elenco dei files di esportazione fino al momento generati. In particolare fornisce i collegamenti ipertestuali, in modo tale che l'utente può facilmente visualizzarne il contenuto.
  - Files dei campioni non esportati. Dopo aver eseguito un'esportazione, il software SEDAMS presenta l'elenco dei files dei campioni non esportati, fino a quel momento generati. In particolare fornisce i collegamenti ipertestuali, in modo tale che l'utente può facilmente visualizzarne il contenuto.

Di seguito è riportata la visualizzazione di una transazione del software SEDAMS con il server provinciale di Verona dell'Agenzia ARPAV:

### **Esportazione dei campioni e delle analisi dal server dbvrlims:**

#### **Periodo di esportazione:**

Data inizio prelevamento: 30/04/2005

Data fine prelevamento: 31/05/2005

#### **Inizializzazione del path:**

Cartella ../export/060531/ esistente

File ../export/060531/eplaf01\_vr.dax esistente

File ../export/060531/eplaf01\_vr.dax rimosso correttamente

Cartella ../not\_export/060531/ esistente

File ../not\_export/060531/sedams\_vr\_err.dat esistente



File ../not\_export/060531/sedams\_vr\_err.dat rimosso correttamente

### Stato della transazione:

Stabilita connessione con il server dbvrlims

Inizio prelevamento ed esportazione dati

Esportato il campione n. 24308 (analisi selezionate = 49)

Esportato il campione n. 24321 (analisi selezionate = 48)

Esportato il campione n. 24387 (analisi selezionate = 2)

Esportato il campione n. 24388 (analisi selezionate = 51)

Esportato il campione n. 24639 (analisi selezionate = 49)

Esportato il campione n. 24640 (analisi selezionate = 49)

....

Esportato il campione n. 24853 (analisi selezionate = 7)

Esportato il campione n. 24988 (analisi selezionate = 1)

Esportato il campione n. 25095 (analisi selezionate = 3)

Il campione n. 25208 non e' stato esportato (analisi selezionate = 0)

Terminata connessione con il server dbvrlims

### Visualizzazione delle statistiche dell'esportazione:

Campioni prelevati: 30

Campioni esportati: 29

Campioni non esportati: 1

Analisi prelevate ed esportate: 392

Numero medio di analisi per campione (prelevato): 13.067

Numero medio di analisi per campione (esportato): 13.517

Dimensione del file di esportazione: 40.08 KB

Dimensione del file dei campioni non esportati: 0.031 KB

Durata della transazione: 0 ore 1 minuti 9 secondi

File di esportazione: eplafr01\_vr.dax creato correttamente

Path file di esportazione:

/home/pdallepezze/public\_html/code\_1\_1/export/060531

File dei campioni non esportati: sedams\_vr\_err.dat creato correttamente

Path file dei campioni non esportati:

/home/pdallepezze/public\_html/code\_1\_1/not\_export/060531

### Files di esportazione:

- [eplafr01\\_pd.dax](#)
- [eplafr01\\_vr.dax](#)

### Files dei campioni non esportati:

- [sedams\\_pd\\_err.dat](#)
- [sedams\\_vr\\_err.dat](#)<sup>7</sup>

---

7. eplafr01\_pd.dax, eplafr01\_vr.dax, sedams\_pd\_err.dat e sedams\_vr\_err.dat sono collegamenti ipertestuali.

### ***3.3. Semplicità d'utilizzo per l'utente***

Uno degli obiettivi primari nella realizzazione del software SEDAMS è stato quello di garantire una tangibile semplicità d'utilizzo del prodotto da parte dell'utente non esperto. Per raggiungere questo scopo, ho costruito il modulo di inserimento dati in modo tale da permettere di compilare i campi richiesti con il solo utilizzo del mouse, ad eccezione dei campi di login e di password, i quali necessitano della tastiera.

Infine, ogni possibile errore, viene segnalato all'utente sul video, assieme ad una breve ed esaustiva descrizione del problema. Nella finestra di aiuto, facilmente raggiungibile con un semplice click del mouse, l'utente è in grado di visualizzare l'errore rilevato, comprenderne la natura e la modalità di intervento per risolverlo.

### ***3.4. Descrizione di un record esportato***

Il file di esportazione è composto da record di 1000 caratteri. I record possono essere di due tipi: record di tipo "01" oppure record di continuazione ("02", "03", etc. ). Ogni record di tipo "01", individua un campione e le sue prime 16 analisi. Nel caso in cui un campione avesse oltre 16 analisi, allora viene anche generato un record di continuazione ("02") per quel campione, nelle quali saranno scritte le successive 31 analisi. Nel caso in cui il campione avesse ancora delle analisi, sarà generato un ulteriore record di continuazione ("03") e così via.

I due tipi di record hanno una struttura diversa. In generale, un record di tipo "01" è composto come di seguito specificato:

1. 463 caratteri sono dedicati ai dati anagrafici del campione (tipo del laboratorio, codice della regione, codice USL, il progressivo del laboratorio, il codice della struttura, etc).
2. 31 caratteri x 16 analisi sono dedicati appunto per le prime 16 analisi del campione.
3. 40 caratteri per la chiusura.

Un record di tipo continuazione è invece composto come segue:

1. 22 caratteri sono dedicati ai dati anagrafici principali del campione in esame.
2. 31 caratteri x 31 analisi sono dedicati per le successive 31 analisi del campione.
3. 16 caratteri per la chiusura.

Si noti che i record sono di lunghezza fissa e quindi, nel caso in cui le analisi di un campione dovessero risultare minori di 16 (per il record di tipo “01”) o minori di 31 (per un generico record di continuazione), si utilizzeranno ugualmente tutti i caratteri fino al completamento del millesimo.

Un esempio di come un campione<sup>8</sup> con 49 analisi viene esportato è fornito di seguito, dove:

- Il testo evidenziato in verde costituisce l'insieme dei dati anagrafici del campione.
- Il testo evidenziato in viola costituisce le analisi del campione.
- Il testo evidenziato in ciano, determina il tipo di record di continuazione.
- Il testo evidenziato in giallo, rappresenta la chiusura del record.

---

8. Si tratta del campione 1671 presente nel database provinciale di Verona. Si ricordi che un record può contenere anche degli spazi. A causa di ciò i record qui presentati non appaiono fedelmente con 1000 caratteri.

P0501200A2003000167101C 190301001

20030109 050023071999 999

R

20030401SI0007G06 NN00 5,31 R0055B01000NN00 R0247A04001NN00 10,53

R0357A27072NN00<0,02 R0369A27072NN00<0,01 R0373A27072NN00<0,01

R0388A27072NN00<0,01 R0410A27072NN00<0,05 R0422A27072NN00<0,01

R0424A27072NN00<0,05 R0425A27072NN00<0,05 R0432A27072NN00<0,05

R0437A27072NN00<0,01 R0440A27072NN00<0,01 R0445A27072NN00<0,01

R0450A27072NN00<0,01 RN01062006

P0501200A20030001671020467A27072NN00<0,01 R0482A27072NN00<0,01

R0485A27072NN00<0,01 R0491A27072NN00<0,01 R0493A27072NN00<0,05

R0507A27072NN00<0,01 R0515A27072NN00<0,02 R0523A27072NN00<0,01

R0530A27072NN00<0,02 R0532A27072NN00<0,01 R0535A27072NN00<0,01

R0542A27072NN00<0,01 R0559A27072NN00<0,1 R0566A27072NN00<0,01

R0567A27072NN00<0,01 R0570A27072NN00<0,05 R0576A27072NN00<0,01

R0577A27072NN00<0,1 R0578A27072NN00<0,01 R0581A27072NN00<0,02

R0594A27072NN00<0,01 R0608A27072NN00<0,01 R0627A27072NN00<0,05

R0671A27072NN00<0,05 R1247A27072NN00<0,01 R1267A27072NN00<0,02

R1309A27072NN00<0,02 R1317A27072NN00<0,01 R1326A27072NN00<0,01

R1330A27072NN00<0,01 R1331A27072NN00<0,01 RN01062006

P0501200A20030001671031334A27072NN00<0,01 R1367A27072NN00<0,01 R0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 0000 00 0000

00 0000 00 N01062006

## ***Capitolo 4. Analisi, progettazione e realizzazione del software***

Questo capitolo illustra *come* il software SEDAMS è stato sviluppato. Al fine di garantire una maggior chiarezza dei contenuti, ho separato la gestione delle attività dalla produzione del software stesso. In questo modo, il lettore è facilmente in grado di comprendere *come* il processo è stato pianificato e controllato e *come* esso è stato eseguito.

I paragrafi 4.1 e 4.2, presentano l'aspetto gestionale ed organizzativo di questo progetto, evidenziando le scelte e le motivazioni del modello di ciclo di vita adottato e la pianificazione delle attività per ciascuna fase di progetto. Accanto ad ogni attività pianificata, si illustrano il rispettivo preventivo e consuntivo delle ore. In questo modo il lettore può immediatamente misurarne l'allontanamento da quanto, in precedenza, era stato previsto.

I paragrafi successivi mostrano, invece, l'aspetto produttivo, introducendo il lettore agli standard adottati, alle problematiche insite nell'esecuzione corretta del processo di sviluppo ed alle attività stesse, dall'analisi dei requisiti al collaudo del sistema.

### ***4.1. Il modello di ciclo di vita***

Il modello di ciclo di vita adottato per il software denominato SEDAMS è quello a cascata. La scelta di questo modello di ciclo di vita è dovuta in particolare a due motivazioni:

- l'importanza di avere una pianificazione chiara e semplice delle attività;
- la rapida comprensione dei requisiti di sistema e del software.

Nella tabella 4.1, è riportata la pianificazione generale del progetto. In particolare

viene presentato come si intende istanziare il modello di ciclo di vita a cascata con riferimento<sup>9</sup> allo standard ISO/IEC 12207:1995.

Il totale della durata di 300 ore di stage, comprensive di attività lavorativa e di stesura di documentazione, sono in accordo con quanto stabilito nel Dipartimento di Matematica Pura e Applicata dell'Università degli Studi di Padova.

Le ore preventivate sono quelle ipotizzate all'inizio del progetto.

<b>N°</b>	<b>Fasi ed attività del processo primario di sviluppo (con riferimento a ISO/IEC 12207:1995<sup>10</sup>)</b>	<b>Preventivo ore</b>	<b>Consuntivo ore</b>
1	<b>Analisi</b>	100	86
	Analisi dei requisiti di sistema (5.3.2).		
	Analisi dei requisiti software (5.3.4).		
2	<b>Progettazione</b>	89	96
	Progettazione architetturale del software (5.3.5).		
	Progettazione di dettaglio del software (5.3.6).		
3	<b>Realizzazione</b>	111	119
	Codifica software (5.3.7).		
	Integrazione software (5.3.8).		
	Collaudo software (5.3.9).		
	Integrazione sistema (5.3.10).		
	Collaudo sistema (5.3.11).		
<b>Totale</b>		<b>300</b>	<b>301</b>

Tabella 4.1. Istanziamento del modello di ciclo di vita a cascata con riferimento allo standard ISO/IEC 12207.

## 4.2. La pianificazione delle attività

Questo paragrafo illustra in forma tabulare le attività previste per ogni fase, la documentazione realizzatasi, i preventivi ed i consuntivi delle ore di stage.

Come presentato nella tabella 4.1 del precedente paragrafo, le fasi di questo progetto sono tre:

9. Lo standard ISO/IEC 12207:1995 è uno standard generico. Esso deve essere istanziato in uno standard aziendale (standard definito) e questo in uno standard di progetto. In questo progetto, mi sono soltanto riferito allo standard ISO/IEC 12207:1995 dal momento che l'Agenzia ARPAV Non dispone di standard definito, dal quale istanziare il mio standard di progetto.

10. La numerazione 5.3 è il processo di sviluppo nello standard ISO/IEC 12207:1995.

- [Analisi](#)
- [Progettazione](#)
- [Realizzazione](#)

### Analisi:

Nella tabella 4.2.1, sono descritte le attività previste per la fase di Analisi.

In questa fase ho compreso che il nucleo dell'intero progetto riguardava la codifica dei singoli campi di un record, ossia un problema fondamentalmente di programmazione.

Tuttavia, ritengo che le fasi di analisi e di progettazione siano state indispensabili per i seguenti tre motivi:

- Garantire la massima efficienza e correttezza nel prodotto software. Mediante un'attenta analisi e progettazione, ho potuto studiare in modo completo la natura del problema ed avanzare diverse soluzioni progettuali.
- Ottenere una chiara ed esaustiva documentazione. In considerazione al punto precedente, lo studio è stato documentato con diagrammi, descrizioni e tabelle, in modo tale che ogni aspetto del software sviluppato sia facilmente comprensibile e reperibile.
- Documentare i processi. Questa considerazione personale si riferisce a come migliorare i processi produttivi stessi, maturando l'importanza di una qualità di processo oltre a quello di prodotto. Ritengo infatti sia necessario e doveroso, per una società evoluta da un punto di vista tecnologico ed informativo, quale la nostra, che si incominci a documentare i singoli processi al fine di massimizzare l'efficienza e ridurre la possibilità di fallimenti.

Tutta la documentazione e le attività previste per questa fase sono state realizzate correttamente. Occorre tuttavia notificare una diminuzione di ore rispetto al

preventivo, pari a 14 ore. Seppure questo sia un errore di pianificazione significativo, non ha comportato un danno per lo svolgimento del progetto, dal momento che le ore preventive sono state sovrastimate. Tali 14 ore, sono state distribuite nelle due prossime fasi.

<b>Fase di Analisi</b>	<b>Preventivo ore</b>	<b>Consuntivo ore</b>
1. Analisi del problema. Studio dell'architettura del database.	14	10
2. Studio sulla reperibilità dei dati. Analisi delle eventuali tabelle da aggiungere nel database.	17	15
3. Stesura del documento <u>Piano di progetto</u> .	15	13
4. Stesura del documento <u>Analisi dei requisiti</u> .	22	18
5. Stesura del documento <u>Piano di qualifica v.1</u>	12	10
6. Inizio stesura del documento <u>Glossario</u> .	1	1
7. Studio del linguaggio PHP.	14	14
8. Verifica	5	5
<b>Totale</b>	<b>100</b>	<b>86</b>

Tabella 4.2.1 Fase di analisi.

### **Progettazione:**

La tabella 4.2.2. descrive le attività previste per la fase di Progettazione.

Tutta la documentazione e le attività previste per questa fase sono state realizzate correttamente. Rispetto alla preventivo, dopo la prima fase, si riscontra un aumento di 3 ore lavorative, dovuto principalmente ad un aumento di ore nella realizzazione del documento Definizione di prodotto. Questa lieve alterazione non ha comportato danni per lo svolgimento della fase successiva.



<b>Fase di Progettazione</b>	<b>Preventivo ore</b>		<b>Consuntivo ore</b>
	<b>Iniziale</b>	<b>Dopo 1 fase</b>	
1. Progettazione del software.	24	25	20
2. Stesura del documento <u>Specifica Tecnica</u> .	25	25	24
3. Stesura del documento <u>Piano di qualifica v. 2.</u>	8	8	9
4. Stesura del documento <u>Definizione di prodotto</u> .	14	14	20
5. Stesura del documento <u>Piano di qualifica v. 3.</u>	8	8	8
6. Verifica	10	13	15
<b>Totale</b>	<b>89</b>	<b>93</b>	<b>96</b>

Tabella 4.2.2 Fase di progettazione.

### **Realizzazione:**

La tabella 4.2.3 descrive le attività previste per la fase di Realizzazione.

Tutta la documentazione pianificata per questa fase è stata prodotta. Rispetto alla pianificazione delle ore prevista dopo la seconda fase, la differenza è minimale (1 ora).

<b>Fase di Realizzazione</b>	<b>Preventivo ore</b>			<b>Consuntivo ore</b>
	<b>Iniziale</b>	<b>Dopo 1 fase</b>	<b>Dopo 2 fase</b>	
1. Codifica.	40	50	57	56
2. Stesura del documento <u>Manuale utente v.1.</u>	8	8	8	8
3. Stesura del documento <u>Piano di qualifica v.4</u>	10	10	8	8
4. Test di unità.	15	15	11	12
5. Test di integrazione.	10	10	8	8
6. Stesura del documento <u>Piano di qualifica v.5.</u>	5	5	5	5
7. Stesura del documento <u>Manuale utente v.2.</u>	7	7	7	8
8. Test di sistema.	8	8	7	7
9. Collaudo col sistema.	8	8	7	7
<b>Totale</b>	<b>111</b>	<b>121</b>	<b>118</b>	<b>119</b>

Tabella 4.2.3 Fase di realizzazione.

### ***4.3. Gli standard di progetto adottati***

Questo paragrafo illustra gli standard adottati nell'analisi, nella progettazione e nella realizzazione del software SEDAMS. Alcuni di questi sono standard ufficialmente noti, come per esempio quelli adottati nella progettazione architettuale, altri istanziati per questo specifico progetto o istanziati da standard definiti.

#### ***4.3.1. Standard di progettazione architettuale***

Nella definizione dell'architettura, si è istanziato il design pattern architettuale denominato three-tier. In particolare, sono presenti tre componenti: l'interfaccia grafica, con la quale l'utente è in grado di interfacciarsi con il sistema, la componente di accesso ai dati e di trattamento dei medesimi ed infine, il database dal quale prelevare i dati.

In questo contesto, lo sviluppo del software SEDAMS costituisce nella realizzazione delle prime due componenti e nell'importazione di alcune tabelle per particolari codifiche ministeriali nel database centrale dell'Agenzia ARPAV.

#### ***4.3.2. Standard di documentazione del codice***

Si è provveduto a documentare appropriatamente ogni intestazione di classe, metodo, funzione, campo dato, variabile globale utilizzata ed il codice interno di funzioni complesse. Si è stabilito di documentare ogni intestazione utilizzando lo stile:

`/** commento */`

in modo tale che, con l'ausilio di appositi strumenti di generazione automatica di documentazione per il linguaggio di programmazione PHP, non servissero ulteriori modifiche.

I campi dati della classe EsportatoreAlimenti, che costituiscono i campi da esportare, sono documentati con la seguente notazione:

CommentoCampoDaEsportare ::= `"/** Campo "` + numero + `": " + tipo + ". " +`  
Descrizione + `(" + byte + "B)." +`

[“(Valore di default).”] + “ \*/”

Per esempio il campo `cod_strut` è documentato come segue:

```
/** Campo 5 : string. Il codice della struttura ARPA (1B). (Valore di default). */
```

#### ***4.3.3. Standard di denominazione di entità e relazioni***

I nomi utilizzati per i campi dati ed i metodi interni al codice sorgente, sono una riduzione intuitiva dei nomi definiti dal Ministero della Salute nel documento SA-99-D1006-015, che costituisce la specifica di un record esportato.

La scelta di usufruire di questa nomenclatura è dovuta a due considerazioni. In primo luogo, il prodotto software SEDAMS non interferisce in alcun modo con il resto dell'architettura dell'Agenzia ARPAV e quindi non sussiste uno standard preesistente per questo contesto. In secondo luogo, tali nomi sono molto esplicativi e simili a quelli adottati internamente dall'Agenzia.

#### ***4.3.4. Standard di programmazione***

Ogni file deve contenere una singola classe, dalla quale prende il nome, questo ai fini di una buona suddivisione logica e di evitare files troppo grandi di difficile comprensione.

Per garantire una buona leggibilità, il codice deve essere ben formattato ed indentato. A tale scopo si è utilizzato lo strumento di indentazione automatica fornito dall'editor stesso. Non è stata imposta una lunghezza massima di riga di codice, dato che la formattazione automatica fornita dall'ambiente di sviluppo, provvede a dividere le righe troppo lunghe.

Eccetto il nome delle classi, per i quali si adotterà lo standard della prima lettera maiuscola, ogni altro nome, sia esso di metodo che di campo dato, sarà in minuscolo.

Eventuali nomi che sono caratterizzati da più parole, saranno suddivisi da un carattere di underscore (“\_”).

Le variabili che saranno inizializzate con i dati prelevati dal database, devono avere il nome della colonna che rappresentano nel database. Ciò è indispensabile

per una rapida consultazione del codice da parte di coloro che dovranno in futuro effettuare la manutenzione.

Tutte le modifiche apportate al codice, devono essere annotate nel registro delle modifiche posto nell'intestazione del file.

#### **4.3.5. Standard di lavoro**

L'Agenzia ARPAV non ha imposto l'utilizzo di particolari strumenti software, eccetto quelli che costituiscono il loro sistema, in particolare Apache, Oracle e TOAD.

Ho quindi deciso di adottare strumenti Open Source, come descritto nel [capitolo 2](#), per evitare qualsiasi licenza commerciale o shareware e per familiarità d'utilizzo.

#### **4.4. Ambientamento**

Durante il primo mese di stage, in concomitanza con le attività pianificate, ho dedicato anche del tempo all'apprendimento di alcuni software utilizzati dall'Agenzia ARPAV e allo studio del linguaggio PHP, inizialmente non noto. Per quanto concerne il primo punto, ho appreso le funzionalità principali del software TOAD 8.5, reperibile nel sito <http://www.toadsoft.com/> . Mediante tale strumento, ho potuto agevolmente individuare le tabelle e i campi richiesti nel mio progetto e occuparmi della gestione di dati mancanti in alcune tabelle (in particolare, il campo rappresentante il codice ministeriale nelle tabelle riguardanti rispettivamente le tecniche analitiche e le unità di misura). Per quanto riguarda, invece, il secondo punto, dopo aver appreso dalla documentazione ufficiale, reperibile nel sito <http://www.php.net/> , gli aspetti fondamentali del linguaggio, ho prestato particolare attenzione allo studio delle funzioni riguardanti la manipolazione delle stringhe e delle date, e la connessione ad un database Oracle. In seguito, l'Agenzia ARPAV ha attivato per il mio account il web server, mediante il quale ho potuto testare le mie pagine PHP.

## 4.5. Analisi dei requisiti

Durante questa fase ho provveduto a ricavare i principali requisiti software e di sistema, classificandoli in requisiti funzionali, di qualità e di interfacciamento. Ai requisiti appartenenti ad ogni classe ho inoltre attribuito una priorità tra le seguenti: obbligatorio, desiderabile e opzionale.

Senza riproporre l'elenco di tutti i requisiti funzionali, per i quali si rimanda al documento di Analisi dei requisiti, presento di seguito i diagrammi Use Case generati, in questa fase.

La figura 4.5.1 illustra le funzioni base del software SEDAMS.

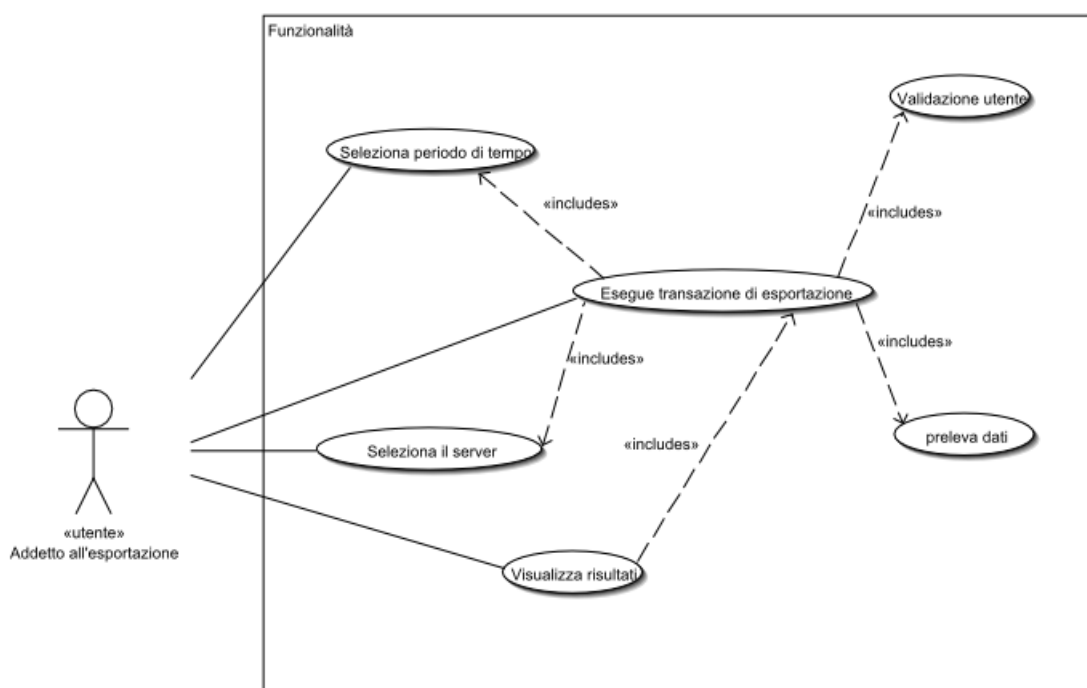


Figura 4.5.1. Diagramma Use Case del funzionamento generale del software SEDAMS.

Da un punto di vista generale, l'utente del software SEDAMS può<sup>11</sup>:

- Selezionare il server provinciale dell'ARPAV al quale vuole connettersi.
- Decidere il periodo di tempo entro il quale vuole esportare i dati riguardanti i campioni e le analisi.

11. In questo diagramma, si fa riferimento alle azioni che l'utente può fare, non a come deve farle.

- Inserire la propria autenticazione con il server provinciale. (Si ricordi che l'utente deve già essere dentro la rete ARPAV).
- Avviare la transazione di esportazione dei dati dal database provinciale.
- Visualizzare il file di esportazione generato.

E' necessario quindi che il software effettui gli opportuni controlli sull'inserimento dei parametri, prelevi ed esporti i dati correttamente.

La figura 4.5.2 illustra il meccanismo di esportazione transazionale del software SEDAMS con maggior dettaglio.

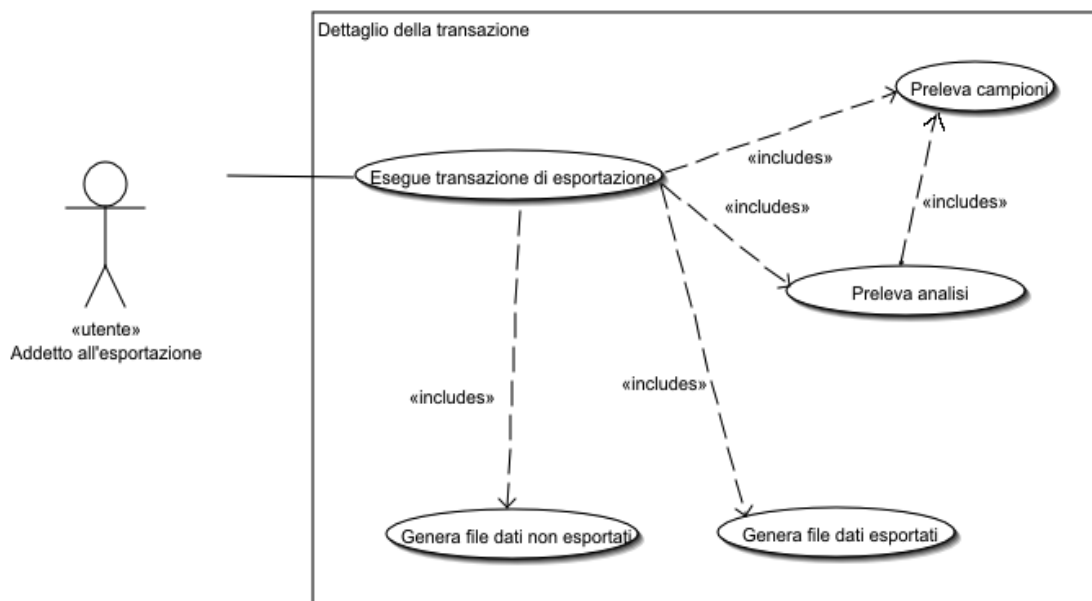


Figura 4.5.2. Diagramma Use Case della transazione.

Dopo aver effettuato tutti i controlli, il software SEDAMS si occuperà di:

- Prelevare i campioni degli alimenti autorizzati.

- Per ogni campione prelevato, prelevare le analisi che sono state autorizzate.
- Esportare i campioni e le analisi con la codifica ministeriale nel file di esportazione.
- Generare un file contenente i campioni non esportati. (Requisito opzionale).

La figura 4.5.3 illustra la generazione e la visualizzazione delle statistiche nel software SEDAMS.

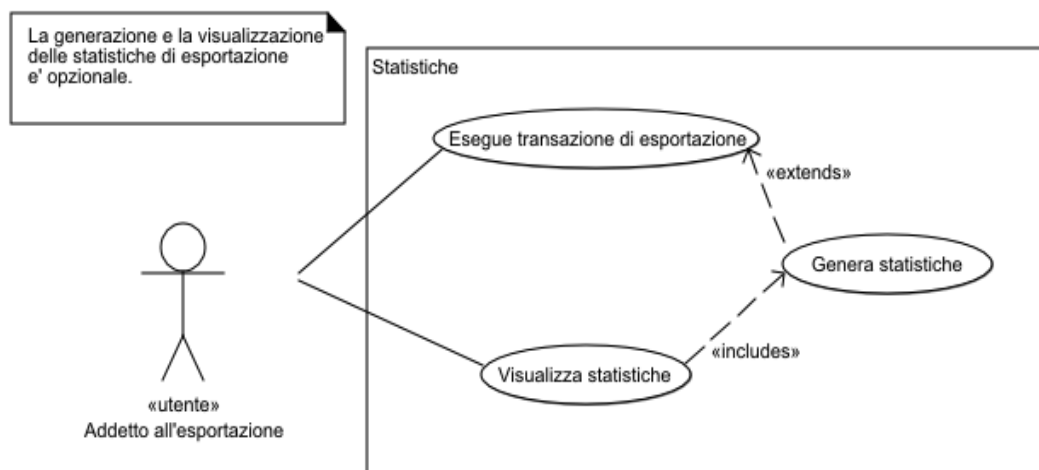


Figura 4.5.3. Diagramma Use Case della generazione e visualizzazione delle statistiche.

Il software, mentre genera il file di esportazione, genera e visualizza a video delle statistiche relative all'esportazione. Le statistiche principali sono quelle specificate nel punto due del [paragrafo 3.2](#).

Mentre avanza la transazione, vengono generate anche le seguenti statistiche relative all'esportazione:

- Il numero dei campioni esportati.
- Il numero dei campioni non esportati.

- Il numero delle analisi esportate.
- La durata dell'esportazione.
- La dimensione del file di esportazione.

Dall'Agenzia ARPAV, sono stati imposti i seguenti due requisiti di qualità:

- Il software SEDAMS deve essere realizzato utilizzando i linguaggi di programmazione PHP e PL/SQL di Oracle 8.1.6.
- Il nome del file contenente i dati da esportare deve essere quello specificato nel documento SA-99-D1006-015.

Per quanto concerne l'interfacciamento con il sistema da parte dell'utente, ho stabilito che ciò avvenga nel modo più semplice ed intuitivo possibile. A tale scopo ho deciso che l'inserimento delle date e la selezione del server provinciale si realizzi utilizzando le combobox, o menù a tendina e che la funzionalità di aiuto sia facilmente raggiungibile e visibile all'utente. Un requisito di interfacciamento importante, imposto dall'Agenzia ARPAV, è che chi accede al software, deve avere i permessi per accedere all'area riservata raggiungibile dal sito dell'Agenzia stessa, ossia deve essere entrato nella rete aziendale. Questo requisito semplifica il meccanismo di autenticazione, in quanto evita un primo accesso al server centrale.

#### ***4.6. Progettazione architetturale***

Nello studio della progettazione architetturale del software SEDAMS ho utilizzato il linguaggio di progettazione UML. In particolare, utilizzando i diagrammi use case, realizzati nella precedente fase di analisi, ho potuto formalizzare una visione compatta e globale del sistema che illustro di seguito adottando i diagrammi di deployment, delle classi, di sequenza e delle collaborazioni.

I server dell'Agenzia ARPAV sono strutturati come descritto in figura 4.6.1, che presenta il diagramma di deployment dell'architettura globale della rete ARPAV.



Nel diagramma si può osservare che l'Agenzia ARPAV è partizionata in sette server regionali caratterizzati dal nome così formalizzato:

database + provincia + LIMS

Per esempio, il database della provincia di Padova è identificato con il nome DBPDLIMS.

I server provinciali sono monitorati e controllati dal server regionale (con sede a Padova), il quale, quindi rappresenta l'Agenzia del Veneto.

Il software SEDAMS sarà installato all'interno del server regionale dell'ARPAV e si conetterà al server provinciale specificato. La motivazione per la quale il software non può interfacciarsi direttamente al server regionale ed esportare tutti i campioni e le analisi alimentari delle sette provincie del Veneto, è dovuta alla mancanza della reperibilità di un consistente numero di dati nel server centrale. In futuro, questa inconsistenza dovrebbe essere colmata.

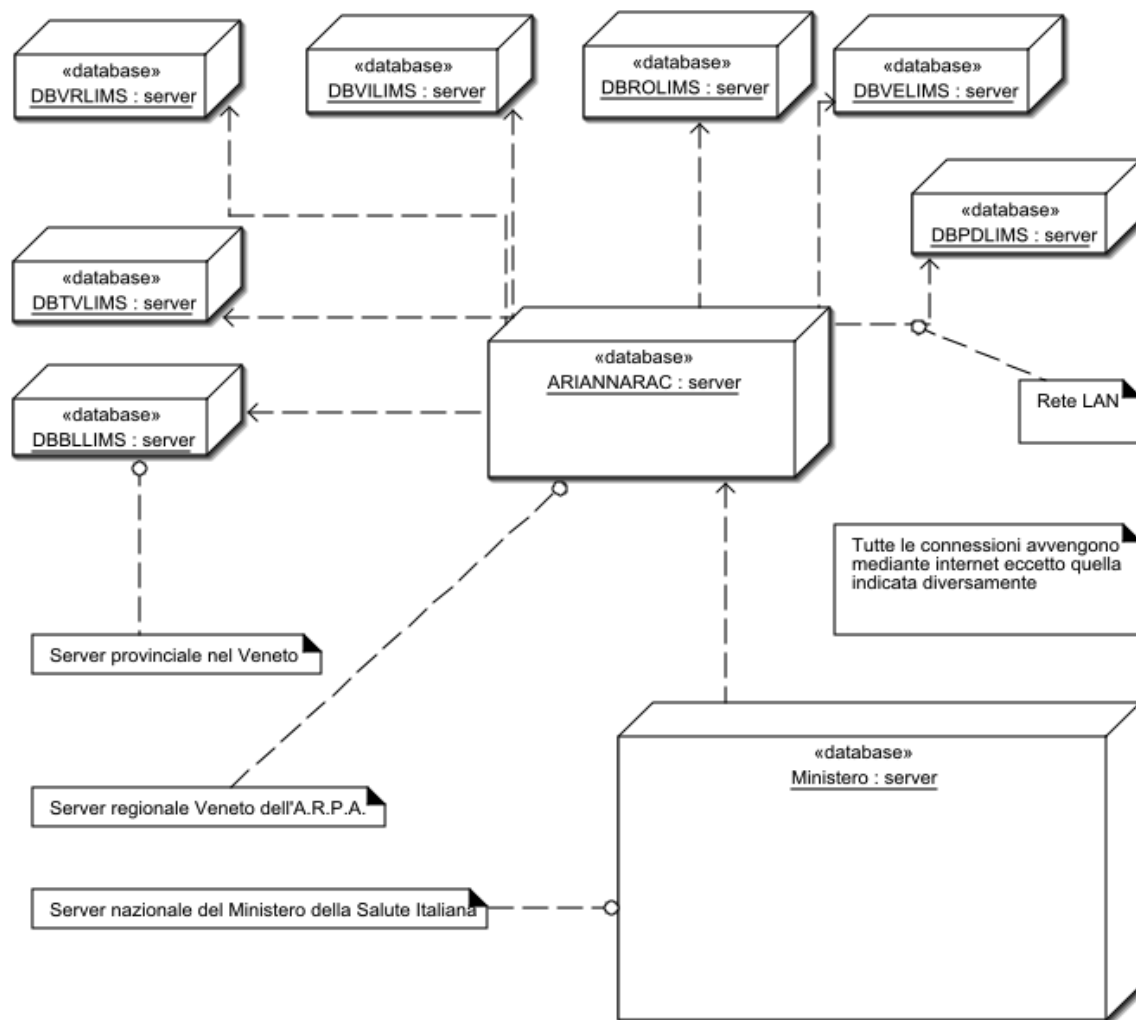


Figura 4.6.1. Diagramma di deployment dei server ARPAV

Nella figura 4.6.2, si presenta il diagramma delle classi dell'architettura astratta del software SEDAMS. Si può osservare immediatamente che si è preso in forte considerazione il design pattern denominato three-tier. In particolare si nota come l'utente si interfaccia alla GUI, la quale richiamerà il servizio che il software garantisce. L'oggetto GUI fornisce un'interfaccia grafica mediante la quale l'utente può utilizzare il servizio del software SEDAMS. Sull'interfaccia grafica l'utente specificherà i parametri descritti nel documento Analisi dei requisiti e potrà visualizzare l'evoluzione della transazione, l'esito e il prodotto. L'oggetto Esportatore genera il file di esportazione dati richiesto, nella codifica ministeriale.

Definisce la transazione nel suo complesso. Fornisce, quindi, anche il servizio di accesso al database. In particolare, tutti i dati richiesti, devono essere prelevati, come specificato nel documento Analisi dei requisiti. L'oggetto DB\_LIMS rappresenta un database di una generica provincia dell'Agenzia ARPAV. Contiene i dati richiesti dal software SEDAMS. Questa componente è già realizzata.

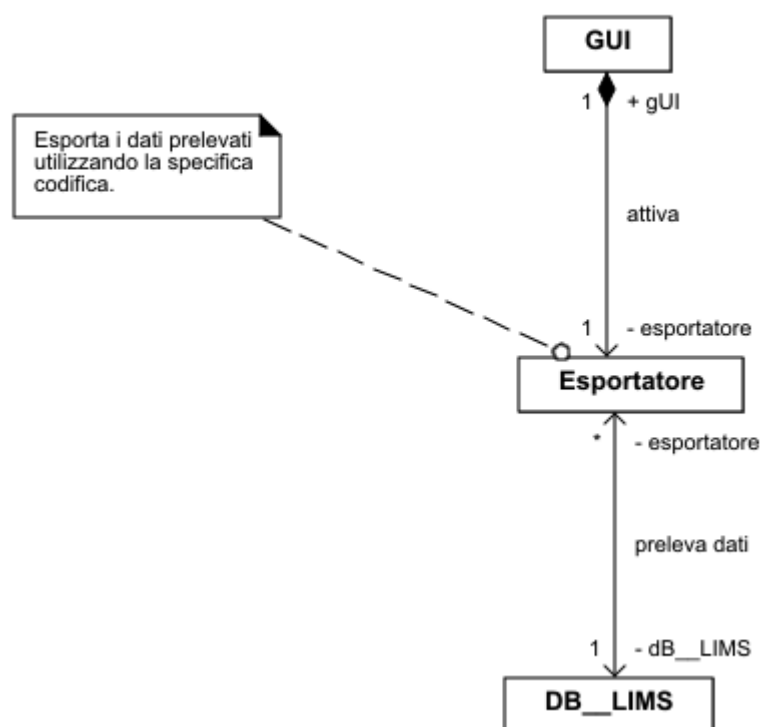


Figura 4.6.2. Diagramma delle classi del software SEDAMS.

Nella figura 4.6.3, si presenta il diagramma di sequenza del software SEDAMS. Fondamentalmente, sono state considerate due possibilità progettuali:

1. effettuare il prelievo e trasporto di tutti i dati necessari, dal server provinciale al server centrale, utilizzando delle tabelle provvisorie. Poi, esportare i dati nella nuova codifica;

2. prelevare tutti i campioni immediatamente e poi, per ciascuno di questi, prelevare le rispettive analisi. Una volta ottenute le analisi, si procederà alla scrittura del campione e delle sue analisi nel file di esportazione senza l'utilizzo di tabelle aggiuntive.

Dopo una discussione, con il tutor aziendale, si è stabilito di scegliere la seconda progettazione per i seguenti due motivi:

1. L'applicazione sarà impiegata periodicamente e non frequentemente.
2. L'Agenzia ARPAV intende nell'imminente futuro, connettere il software direttamente nel server regionale e da questo, prelevare tutti i dati dalle sette province del Veneto in una sola volta. Con la seconda progettazione e con queste intenzioni, si risparmieranno sicuramente tempo e spazio, in quanto non sarà necessario l'utilizzo di tabelle intermedie e la riscrittura dei dati.

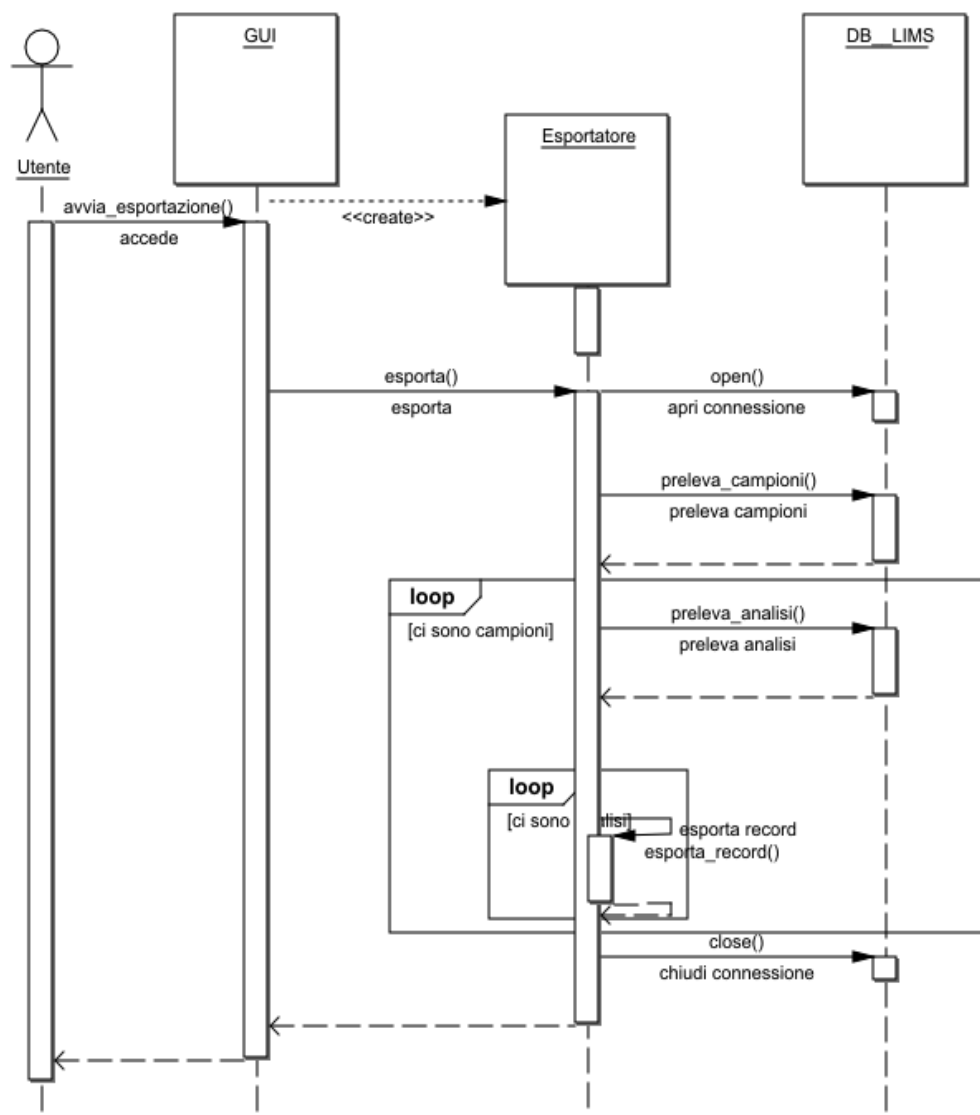


Figura 4.6.3. Diagramma di sequenza del software SEDAMS.

Nella figura 4.6.4, è presentata la vista rappresentante la collaborazione tra gli oggetti. Tale diagramma è stato costruito a partire dal precedente diagramma UML di sequenza.

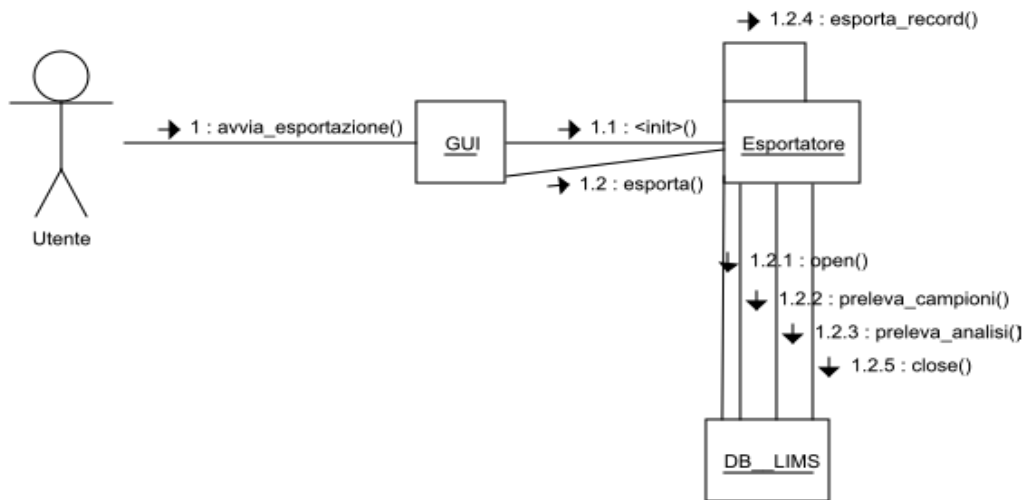


Figura 4.6.4. Diagramma di collaborazione del software SEDAMS.

Ricercando nelle entità del database dell'Agenzia ARPAV, la posizione specifica dei campi richiesti dal Ministero della Salute, ho notato che alcuni di essi non sono gestiti dall'Agenzia ARPAV,

E' stato quindi concordato, in una riunione interna che, mentre per alcuni campi la gestione sarebbe stata affidata ai laboratori LIMS delle sette province, per i rimanenti l'incarico di gestirli sarebbe stato affidato al sottoscritto. In particolare, mi sono occupato di gestire i seguenti campi:

- laboratorio\_tipo, codice\_regione, codice\_usl, progressivo\_laboratorio;
- tecniche\_analitiche;
- unità\_misura.

Per il campo tecniche\_analitiche ho dovuto ricorrere all'inserimento di una tabella ausiliaria nel database ARPAV, mentre per il campo unità\_misura, è stato sufficiente riempire una colonna che inizialmente era vuota.

Il software è concepito per esportare i dati che sono stati autorizzati nel periodo di 1 anno. Ho ritenuto importante effettuare e documentare una stima delle misure più rilevanti che interessano il software SEDAMS. Il numero medio dei campioni per anno e il numero medio delle analisi per campione mi sono stati forniti dall'Agenzia stessa. Nella tabella 4.6.1, si forniscono delle stime di misurazione:

<b>Oggetto</b>	<b>Misura stimata</b>
Numero medio di campioni / anno	1300
Numero medio di analisi per campione	40
Campi da prelevare per 1 campione	26
Campi da prelevare per 1 analisi di 1 campione	7
Numero medio di informazioni / anno	52000 (= 1300 campioni * 40 analisi)
Grandezza file di esportazione	104 MB (codifica ASCII) (= 52000 * 1 B * (2000 celle per informazione))
Durata della transazione prevista	45 minuti

Tabella 4.6.1. Stima delle misurazioni per il software SEDAMS.

#### **4.7. Progettazione di dettaglio**

Dopo aver presentato la progettazione architeturale, illustro in questo paragrafo la progettazione di dettaglio, che specifica effettivamente la struttura definitiva del software SEDAMS.

La struttura dei files è presentata mediante il diagramma UML delle componenti nella figura 4.7.1. Eccetto i moduli Esportatore e EsportatoreAlimenti, per i quali si è adottata una progettazione ad oggetti, tutti gli altri sono di fatto progettati e realizzati utilizzando procedure, dal momento che rappresentano pagine web con al più un breve algoritmo di stampa.

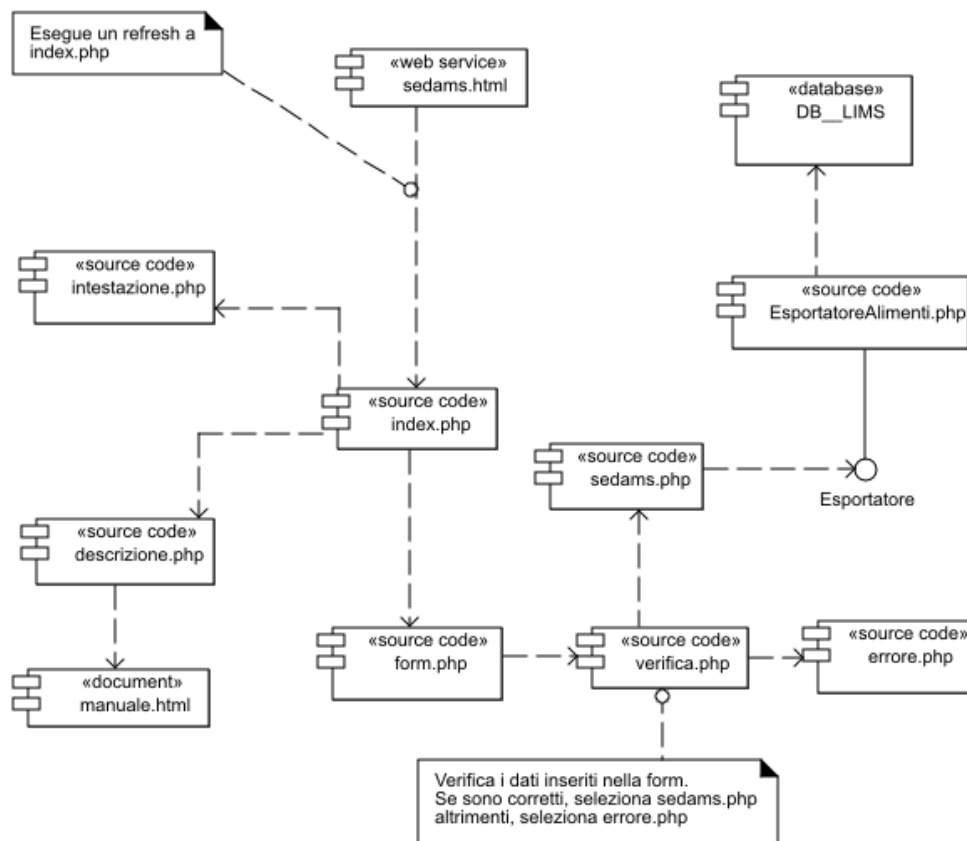


Figura 4.7.1. Diagramma delle componenti che illustra la struttura dei files.

L'interfaccia grafica è realizzata con i linguaggi HTML e PHP ed è suddivisa in diversi frame che costituiranno l'area di inserimento dati (form.php), l'area di visualizzazione dello stato attuale della transazione (sedams.php) e l'area di descrizione del prodotto (descrizione.php). Da quest'ultima è possibile visualizzare il manuale utente completo (manuale.html).



L'interfaccia grafica utente tratta due tipi di dati:

1. Parametri inseriti dall'utente:
  - 1.1. Login dell'utente.
  - 1.2. Password dell'utente.
  - 1.3. Server provinciale al quale ci si intende connettere.
  - 1.4. La data a partire dalla quale si intende effettuare l'esportazione.
  - 1.5. La data di fine esportazione.
  - 1.6. Un bottone di avviamento dell'esportazione.
2. Il link al file di esportazione prodotto. Il nome del file da esportare al Ministero della Salute, secondo il documento SA-99-D1006-015 è eplafr01.dax. I files provinciali prodotti avranno per cui il nome:  

"eplafr01\_" + SiglaProvinciaVeneta + ".dax".

Nella figura 4.7.2, è presentato il diagramma UML delle classi della logica applicativa. Di seguito, vengono analizzati i singoli moduli, rispettivamente Esportatore e EsportatoreAlimenti.



Figura 4.7.2. Diagramma delle classi che illustra l'architettura di dettaglio.

L'interfaccia Esportatore definisce le specifiche generali di un generico esportatore dati. L'interfaccia grafica del software SEDAMS si interfaccia direttamente con questo modulo.

Il modulo EsportatoreAlimenti è una classe che implementa l'interfaccia Esportatore. Lo scopo è quello di generare il file di esportazione dati richiesto, nella codifica ministeriale. Questa classe definisce la transazione nel suo complesso

Le tabella 4.7.1 e 4.7.2 riportano rispettivamente le descrizioni dei campi dati e dei metodi della classe EsportatoreAlimenti.

<b>Nome</b>	<b>Descrizione</b>
<b>Comandi SQL</b>	
sql_preleva_campioni	Stringa di SELECT per il prelevamento dei campioni di tipo ALIMENTI autorizzati.
sql_preleva_analisi	Stringa di SELECT per il prelevamento delle analisi di un campione di tipo ALIMENTI autorizzato.
<b>Periodo di esportazione</b>	
data_ini	Definisce la data iniziale a partire dalla quale si intende esportare.
data_fin	Definisce la data finale entro la quale si intende esportare.
<b>Parametri di connessione</b>	
server	Il nome del server a cui ci si intende connettere.
login	La login dell'utente.
password	La password dell'utente.
<b>Campi dei campioni</b>	
cod_tipi_lab	Un array contenente i sette codici pmp delle province della regione Veneto.
cod_pmp	Il codice della provincia della regione veneto alla quale ci si connette. (8B)
cod_strut	Il Codice della struttura ARPA (1B). (Valore di default).
anno_camp	L'anno di revisione del campione (4B).
prog_camp	Il progressivo del campione (6B).
grado_istanza	Grado di istanza (1B). (Valore di default).
tipo_record	Tipo di record (2B). (Campo del Ministero della Salute).
tipo_esame	Il tipo dell'esame che è stato fatto sul campione. L'esame può essere di tipo chimico (C), biologico (B) o fisico (F).
cod_alim_appart	Il codice dell'alimento (9B).

<b>Nome</b>	<b>Descrizione</b>
den_alim	La denominazione dell'alimento (60B).
data_prel	Una stringa contenente l'anno, il mese e il giorno del prelievo del campione nel formato YYYYMMDD(8B).
motivo_prel	Il motivo del prelievo (2B). (Valore di default).
cod_filiera_alim	Il codice della filiera alimentare (1B). (Valore di default).
cod_reg_prel	Il codice della regione nella quale è stato effettuato il prelievo (3B).
cod_prov_prel	Il codice della provincia nella quale è stato effettuato il prelievo (3B).
cod_com_prel	Il codice del comune nella quale è stato effettuato il prelievo (3B).
org_prelevatore	Il codice dell'organismo che ha effettuato il prelievo del campione (3B).
org_prelevatore_utente	Organismo prelevatore utente (3B). (Valore di default).
cod_fisc_prod	Il codice fiscale del produttore (16B). (Valore di default).
den_prod	La denominazione del produttore (40B).
reg_naz_prod	Il codice della regione del produttore se questo è in Italia, altrimenti il codice della nazione
cod_fisc_dist	Il codice fiscale del distributore (16B). (Valore di default).
den_dist	La denominazione del distributore (40B).
mod_conser	Il codice che rappresenta la modalità di conservazione del campione (2B).
tipo_conten	Il codice che rappresenta il tipo di contenitore utilizzato (2B).
lab_sec_istanza	La seconda istanza di laboratorio (3B). (Valore di default).
esito_globale	L'esito globale del campione (1B). (Valore di default).
note	Note di considerazioni (210B). (Attualmente valore di default).
data_fine_analisi	Stringa contenente l'anno, il mese ed il giorno in cui l'analisi è stata completata nel formato YYYYMMDD (8B).
controllo_ufficiale	Afferma se sull'analisi è stato eseguito un controllo ufficiale (2B). (Valore di default).
<b>Campi delle analisi</b>	
cod_det	Il codice di determinazione (4B).
prog_det	Il progressivo di determinazione (1B).
tecn_anal	Il codice della tecnica analitica impiegata (2B).
unit_mis	Il codice dell'unità di misura impiegata (3B).
flag_det	Il flag di determinazione (1B). (Valore di default).
flag_accr	Il flag di accreditamento (1B). (Valore di default).
unit_camp	L'unità campionaria (2B). (Valore di default).
segno	Il segno (1B).

<b>Nome</b>	<b>Descrizione</b>
ris_sing_det	Il risultato della singola determinazione (15B).
esito_analisi	L'esito dell'analisi (1B).
<b>Campi di chiusura</b>	
program_minister	Afferma se questo è un programma ministeriale (1B). (Valore di default).
data_inserimento	La data attuale (8B) (DDMMYYYY).
filler_rec_01	Stringa di blank di chiusura per i record di tipo 01 (32B). (Valore di default).
filler_rec_cont	Stringa di blank di chiusura per i record di continuazione (8B). (Valore di default).

Tabella 4.7.1 Campi dati della classe EsportatoreAlimenti.

<b>Nome</b>	<b>Descrizione</b>
EsportatoreAlimenti(\$data_inizio,\$data_fine, \$server, \$login, \$password)	Inizializza i campi dati ricevuti e quelli di default secondo il documento <u>Codifica Esportazione</u> .
esporta(\$filename, \$filename_err)	Rappresenta il motore dell'esportatore. Si occupa di prelevare i campioni, le analisi di questi, convertire i valori ed esportare i dati. I campioni e le analisi esportati correttamente saranno scritte nel file \$filename, mentre i campioni non esportati nel file \$filename_err.
esporta_campione_01(\$sample_number, \$date_completed, \$av_un_funzionale, \$av_codice_alimento, \$description, \$sampled_data, \$av_punto_prelievo, \$av_prelevatore, \$av_prodotto, \$av_distributore, \$av_conservazione, \$av_contenitore, \$date_reviewed)	Ritorna la stringa ottenuta dalla concatenazione di tutti i campi del campione da esportare (per il record di tipo 01).
esporta_campione_cont()	Ritorna la stringa ottenuta dalla concatenazione di tutti i campi del campione da esportare (per il record di continuazione).
esporta_analisi(\$cod_det, \$av_ministero, \$entry_qualifier, \$numerical_entry, \$in_spec)	Ritorna la stringa ottenuta dalla concatenazione dei campi dell'analisi del campione selezionato.
esporta_chiusura()	Ritorna la stringa ottenuta dalla concatenazione dei campi appartenenti alla chiusura. Il metodo stabilisce se deve ritornare una chiusura per un record di tipo 01 o di continuazione, in base alla variabile \$tipo_record.
esporta_riempimento(\$righe_analisi_report)	Ritorna una stringa contenente il riempimento del record. Dipende dalla variabile \$tipo_record.
setta_stringa_blank(\$len)	Ritorna una stringa di blank lunga quanto specificato dal parametro passato, che dovrà essere un intero non negativo.

<b>Nome</b>	<b>Descrizione</b>
setta_stringa_zeri(\$len)	Ritorna una stringa di zeri lunga quanto specificato dal parametro passato, che dovrà essere un intero non negativo.
cambia_server(\$server, \$login, \$password)	Cambia il server di connessione.
set_cod_pmp()	Setta il campo cod_pmp, secondo il nome server di connessione. In particolare, vengono settati: lab_tipo + cod_reg + cod_usl + prog_lab. (8B). Tale tabella è già fornita dal Ministero della Salute (Descrizioni PMP).
set_anno_camp(\$date_completed)	Setta il campo anno_camp secondo la codifica.
set_prog_camp(\$sample_number)	Setta il progressivo del campione. Se il sample_number ha più di 6 cifre, si tronca alla settima cifra a sinistra. Il riempimento deve essere fatto con zeri a sinistra.
set_esame_tipo(\$av_un_funzionale)	Setta il campo esame_tipo. L'esame può essere chimico (C), fisico (F) o biologico (B).
set_cod_alim_appart (\$av_codice_alimento)	Setta il campo cod_alim_appart
set_den_alim(\$description)	Setta il campo den_alim.
data_prel	Setta la data del prelievo del campione. Se questa è nulla, setta la data con zeri.
set_cod_localita_prel(\$av_punto_prelievo)	Setta il codice della regione, della provincia e del comune nel quale è stato effettuato il prelievo.
set_org_prelevatore(\$av_prelevatore)	Setta il campo org_prelevatore.
set_den_prod(\$av_prodotto)	Setta il campo den_prod.
set_reg_naz_prod()	Setta il campo reg_naz_prod.
set_den_dist(\$av_distributore)	Setta il campo den_dist.
set_mod_conserv(\$av_conservazione)	Setta il campo mod_conserv.
set_tipo_conten(\$av_contenitore)	Setta il campo tipo_conten.
set_note()	Setta il campo note.
set_data_fine_analisi(\$date_reviewed)	Setta il campo data_fine_analisi.
set_cod_det(\$av_ministero)	Setta il campo cod_det.
set_prog_det()	Setta il campo prog_det.
set_tecn_anal()	Setta il campo tecn_anal.
set_unit_mis()	Setta il campo unit_mis.
set_segno(\$entry_qualifier)	Setta il campo segno.
set_ris_sing_det(\$numerical_entry)	Setta il campo ris_sing_det.
set_esito_analisi(\$in_spec)	Setta il campo esito_analisi.

Tabella 4.7.2. I metodi della classe EsportatoreAlimenti.

## 4.8. Realizzazione

Il codice sorgente comprende i files per l'interfaccia grafica, per l'esportazione dei dati, per la documentazione in linea e le immagini utilizzate.

Di seguito si fornisce l'elenco dei files consegnati:

1. Cartella "src".
  - 1.1. File "manuale.html".
  - 1.1. File "sedams.html".
  - 1.2. Cartella "sedams".
    - 1.2.1. File "descrizione.php".
    - 1.2.2. File "errore.php".
    - 1.2.3. File "Esportatore.php".
    - 1.2.4. File "EsportatoreAlimenti.php".
    - 1.2.5. File "form.php".
    - 1.2.6. File "index.php".
    - 1.2.7. File "intestazione.php".
    - 1.2.8. File "sedams.php".
    - 1.2.9. File "verifica.php".
    - 1.2.10. File "stile.css".
    - 1.2.11. Cartella "images".
      - 1.2.11.1. File "nebulosa.jpg".
      - 1.2.11.2. File "stars.jpg".
      - 1.2.11.3. File "tellus\_luna.jpg".

Il software crea due cartelle denominate */export* e */not\_export* nello stesso path in cui è situato il file *sedams.html*. In ognuna di queste due cartelle sarà creata un'ulteriore cartella */YYYYMMDD* dove YYYY, MM e DD rappresentano rispettivamente l'anno, il mese ed il giorno in cui si esegue l'esportazione. All'interno di queste cartelle saranno successivamente posti il file di esportazione

(nel path */export/YYYYMMDD*) ed il file dei campioni non esportati (nel path */not\_export/YYYYMMDD*). Il nome di quest'ultimo file è del tipo:

"sedams\_" + SiglaProvinciaVeneta + "\_err.dat".

#### 4.9. Verifica

Poiché nella realizzazione di questo progetto ero solo, è stato molto importante un confronto periodico sulle attività svolte attraverso un colloquio frequente con il tutor e la realizzazione di documentazione.

A tale scopo ho adottato il ciclo PDCA al prodotto nel seguente modo:

- **Pianificazione.** Nei documenti Piano di progetto e Piano di qualifica, ho documentato appositamente tutta la pianificazione dell'intero progetto. Il Piano di progetto contiene tutte le informazioni indispensabili per la gestione lavorativa dello stagista. Per ogni attività da svolgere ho definito *cosa* e *come* ognuna di esse deve essere compiuta. Nel Piano di qualifica, invece, ho documentato la metodologia di verifica di ogni fase, dei documenti e del codice.
- **Esecuzione.** Comporta l'esecuzione delle attività previste nel piano di progetto per il processo software coinvolto. Per esempio ricerca ed analisi dei requisiti e stesura del documento.
- **Controllo.** Nel caso di verifica, ho controllato che ogni post-condizione, pianificata nei documenti Piano di progetto e Piano di qualifica sia stata completamente soddisfatta nel prodotto software ottenuto con quello specifico processo. Nel caso di validazione, ho utilizzato i documenti realizzati precedentemente, controllando il soddisfacimento di tutti i requisiti. Tutti i controlli sono stati pianificati e gli esiti ottenuti sono stati registrati per testimoniare l'uscita corretta da quel processo o da quella fase.
- **Azione.** A seconda che il controllo riveli la presenza di fallimento, errore o guasto, ho stabilito di intervenire nel documento realizzato e ripetere il ciclo



PDCA.

Per garantire la qualità di processo, si ritiene necessario documentare anche l'evoluzione del processo stesso, in particolar modo riferendone la qualità del suo sviluppo. Ho cercato, quindi, di applicare il ciclo PDCA ai processi stessi come specificato di seguito:

- **Pianificazione.** Ho cercato di pianificare come il processo dovrà svolgersi.
- **Esecuzione.** Consiste nella documentazione dello svolgimento del processo.
- **Controllo.** Ho controllato l'esito del processo con quanto atteso nella pianificazione del processo stesso.
- **Azione.** Ho documentato come il processo può essere migliorato, in riferimento al controllo.

Ogni settimana, ho sospeso l'avanzamento del progetto assumendo il ruolo di verificatore, impegnandomi a controllare l'evoluzione del processo e del prodotto.

## *Capitolo 5. Gestione della qualità*

L'obiettivo a cui ho attribuito maggiore importanza durante il processo di sviluppo è stato quello di garantire un prodotto qualitativamente accettabile. A tal fine, ho compreso la necessità di apportare una gestione della qualità ad ogni attività. Infatti, se ogni attività è ben correlata alle altre ed ognuna dispone di un'efficiente gestione della qualità, allora il processo che le ingloba avrà sicuramente una qualità soddisfacente.

All'inizio di ogni attività, ho per cui intervistato il tutor Paolo Zambotto<sup>12</sup> e redatto un resoconto delle sue aspettative. Da questo documento interno, in primo luogo ho pianificato i compiti necessari, in secondo luogo ho tentato di stabilire una scala di misura mediante la quale poter misurare la qualità del mio prodotto.

La più importante scala di misura realizzata è quella che rappresenta il soddisfacimento dei requisiti. Tale scala è stata realizzata in maniera soggettiva, facendo riferimento come potrei io stesso giudicare il prodotto se fossi un possibile cliente. Ho individuato sette valori possibili:

1. Assente. Nessun requisito è soddisfatto.
2. Incompleto. Soltanto alcuni requisiti sono soddisfatti, anche se non si sa ancora nulla sulla loro effettiva correttezza.
3. Realizzazione del servizio (non testato). L'esportatore è completato e risponde a tutti i requisiti software.
4. Servizio testato. L'esportatore è completato e funziona correttamente.
5. SEDAMS completo (non user-friendly). Il software è munito di interfaccia grafica, funziona ed è corretto. Tuttavia non è di facile utilizzo. L'utente deve ricorrere per ogni inserimento dati alla tastiera.
6. SEDAMS completo (user-friendly). Il software è di facile utilizzo anche da parte di utenti non esperti. Molte funzionalità sono richiamabili mediante il

---

12. In questo contesto è il committente.

solo utilizzo del mouse.

7. SEDAMS con visualizzazione grafica e chiara della transazione. Il software illustra graficamente all'utente l'evoluzione della transazione e mediante opportuni grafici le statistiche.

Associando questa scala di misura all'asse temporale, ho ottenuto il grafico in figura 5.1.



Figura 5.1. Misurazione del soddisfacimento dei requisiti.

In questo grafico, si può osservare come sono arrivato e con che tempi al livello 6 della mia scala di qualità. Il settimo livello non è stato raggiunto per motivi di mancanza di tempo.

## *Conclusioni*

Al termine dell'esperienza di stage, ritengo che gli obiettivi inizialmente prefissati siano stati raggiunti con successo. Come si può osservare nella pianificazione, non sono stati segnalati rallentamenti significativi ed ogni fase ha avuto esito positivo.

L'esperienza presso l'Agenzia ARPAV è stata importante per diverse ragioni. Innanzitutto, ho potuto consolidare ed integrare le conoscenze apprese soprattutto nei corsi di Ingegneria del Software e Basi di Dati e contestualizzarle in un ambiente reale. Molta attenzione è stata dedicata alla documentazione adottando costantemente il ciclo PDCA, alla realizzazione dei diagrammi ed allo studio di diversi scenari. I corsi di Basi di Dati mi sono stati utili per comprendere le nozioni necessarie a gestire i database, in particolare la programmazione di procedure in linguaggio PL/SQL.

Buona parte del tempo inoltre, è stata investita nella progettazione di un software che fosse riutilizzabile in futuro, sia in caso di modifica del sistema di esportazione dati, che per riutilizzare parte del codice per creare soluzioni a progetti aventi parti in comune. Al termine di ogni fase, ho inoltre redatto un breve paragrafo che descrivesse i problemi riscontrati per ogni attività e come si sarebbero potuti evitare o migliorare, in modo tale che gli stessi processi, potessero essere volendo riutilizzati in contesti simili.

Infine, ho cercato di ottimizzare il processo di sviluppo ponendo una considerevole attenzione a tutti quegli aspetti che, in qualche modo, hanno comportato inefficienza e non praticità nei progetti dei corsi sopra citati. In particolare, ho tentato di rendere più efficienti i processi di documentazione e di versionamento e configurazione preferendo l'uso di tabelle con descrizioni succinte e significative e adottando strumenti automatici nell'editor di testo (inserimento automatico del numero di pagina, l'indice con collegamenti ipertestuali, versione e data aggiornabili automaticamente etc.). Ciò ha garantito in primo luogo la

realizzazione di documentazione chiara, esaustiva e di facile consultazione ed in secondo luogo un controllo più rapido e sicuro.

Il codice sorgente del software SEDAMS consta di circa 2200 righe di codice scritte nei linguaggi PHP, HTML, SQL e CSS, mentre sono state redatte circa 110 pagine in documenti. In media, un file di esportazione contenente i campioni e le analisi prelevate nell'arco di un anno è circa di 4 MB e la transazione impiega nel suo complesso circa 40 minuti, prelevando i campioni e le analisi dal server di Verona.

L'attività complessiva di stage è durata un mese e tre settimane. Durante questo periodo ho potuto lavorare presso un'azienda reale, con personale disponibile e altamente competente. Ciò mi ha permesso di prendere maggiormente coscienza di cosa significhi il mondo del lavoro aziendale e come le persone cooperano tra loro per migliorare il processo produttivo. L'ARPAV mi ha offerto un ambiente nuovo rispetto alle mie esperienze precedenti in cui interagiscono strutture di ambiti diversi, banche dati di vaste dimensioni e sistemi informativi complessi e vari. L'Agenzia, ha messo a disposizione per la mia attività, una postazione di lavoro con strumentazione efficiente e moderna. Per quanto riguarda la gestione di alcune informazioni mancanti da parte dei sette laboratori provinciali LIMS, tutto è stato svolto il più rapidamente possibile in modo da non compromettere o ritardare il mio operato. Inoltre, per ogni dubbio riguardante la struttura del database ARPAV, il tutor aziendale Paolo Zambotto ed il personale sono sempre stati disponibili. Grazie ai loro immediati chiarimenti, ho potuto completare la mia attività di stage entro i tempi previsti. Il tutor, dall'altra parte, mi ha sempre lasciato lavorare in autonomia, rispettando le mie scelte progettuali ed informandosi costantemente sullo stato di avanzamento.

Nello svolgimento di questo progetto, ho potuto apprendere il linguaggio di programmazione PHP, il quale si è rilevato adeguato per la realizzazione di un'applicazione web. Grazie alle innumerevoli funzionalità già offerte dal linguaggio stesso, ho per esempio evitato di affrontare il problema della correttezza delle date immesse, agevolando così lo sviluppo ed i test del software SEDAMS.

## *Appendice*

Per un eventuale consultazione di maggior dettaglio del software SEDAMS, di seguito ho aggiunto i documenti formali esterni da me realizzati durante questo periodo di stage. Per evitare ridondanza, il glossario è stato integrato con quello di questa tesi. I documenti prodotti sono:

- [Piano di progetto](#)
- [Piano di qualifica](#)
- [Analisi dei requisiti](#)
- [Specifica tecnica](#)
- [Definizione di prodotto](#)
- [Manuale utente](#)

Seppure citati, non sono stati allegati il codice sorgente in quanto, al termine della mia attività di stage, è diventato proprietà dell'Agenzia ARPAV ed il documento formale interno Codifica esportazione, perché cita esplicitamente i nomi dei campi e delle tabelle, del database ARPAV, che dovranno essere prelevati. Per garantire una certa riservatezza di parte della struttura del database ARPAV, ho preferito omettere questi documenti.

## *Piano di progetto*

**Nome del prodotto:**

Sistema di Esportazione Dati Alimenti per il Ministero della Salute  
(SEDAMS).

**Data:**

10/06/06

**Versione:**

1.0

**Stato del documento:**

Formale esterno

**Redazione:**

Piero Dalle Pezze

**Revisione:**

Piero Dalle Pezze    Padova, lì 10/06/06    Firma .....

**Approvazione:**

Paolo Zambotto    Padova, lì 10/06/06    Firma .....

**Lista di distribuzione:**

1. Paolo Zambotto
2. Piero Dalle Pezze

## **Registro delle modifiche:**

v 1.0 09/06/06

Revisione formale del documento.

v 0.12 08/06/06

Report della fase di realizzazione.

v 0.11 07/06/06

Registrazione consuntivo totale per la fase di realizzazione.

v 0.10 01/06/06

Registrazione consuntivo parziale per la fase di realizzazione.

v 0.9 24/05/06

Registrazione del consuntivo finale per la fase di progettazione.

Registrazione consuntivo per la codifica nella fase di realizzazione.

v 0.8 10/05/06

Registrazione del consuntivo intermedio per la fase di progettazione.

v 0.7 04/05/06

Registrazione del consuntivo parziale per la fase di progettazione.

v 0.6 28/04/06

Revisione formale del documento.

v 0.5 26/04/06

Riformulazione del capitolo 2. Registro del consuntivo di prima fase.

v 0.4 21/04/06

Integrazione del paragrafo 2.2.

v 0.3 20/04/06

Registrazione consuntivo parziale della prima fase.

v 0.2 19/04/06

Modifica della tabella 2.2. (Aggiunta redazione Manuale utente v1-2).

Aggiunti paragrafi 3, 4 e 5.

v 0.1 18/04/06

Creazione di questo documento.



## Sommario:

Questo documento contiene la pianificazione dello stage svolto dallo studente Dalle Pezze Piero, presso ARPAV.

## Indice:

1. Introduzione	58
1.1. Scopo del documento	58
1.2. Scopo del prodotto	58
1.3. Glossario	58
2. Pianificazione	59
2.1. Attività di stage	59
2.2. Dettaglio delle fasi del ciclo di vita	60
2.2.1. Analisi	60
2.2.2. Progettazione	62
2.2.3. Realizzazione	64
3. Gestione dei rischi di progetto	67
3.1. Piano di progetto inefficace	67
3.2. Gestione del piano di progetto inadeguata	67
4. Gestione della qualità del piano di progetto	68
5. Gestione del piano di progetto	69

# ***Capitolo 1. Introduzione***

Di seguito sono riportate le informazioni generali riguardanti questo documento.

## ***1.1. Scopo del documento***

Lo scopo di questo documento è documentare l'organizzazione e la pianificazione delle attività dello studente Dalle Pezze Piero nel progetto stage denominato *SEDAMS*. Ciò è necessario per poter realizzare un prodotto affidabile, semplice all'utilizzatore, completo rispetto ai requisiti stabiliti, portabile, possibilmente riusabile in differenti contesti e facilmente manutenibile. Per raggiungere ciò, è stato compresa la necessità di una pianificazione delle attività che risulti ben gestita, efficiente e controllata.

## ***1.2. Scopo del prodotto***

Lo scopo del prodotto è quello di esportare dati nella codifica richiesta dal Ministero della Salute specificata nel documento *SA-99-D1006-015*.

## ***1.3. Glossario***

Il glossario viene fornito come documento allegato ed è unico per tutta la documentazione.

## 2. Pianificazione

Questo capitolo contiene l'intera pianificazione delle attività dello stagista Dalle Pezze Piero. Nella tabella 2.1 sono riportati i dati generali di contratto.

<b>Nominativo</b>	Dalle Pezze Piero
<b>Matricola</b>	486228-sin
<b>Data inizio</b>	18/04/06
<b>Durata stage</b>	300 ore
<b>Mail</b>	pdallep@studenti.math.unipd.it
<b>Firma</b>	

Tabella 2.1. Informazioni contrattuali.

### 2.1. Attività di stage

Il modello di ciclo di vita adottato per il software denominato *SEDAMS* è quello a cascata. La scelta di questo modello di ciclo di vita si è riflettuta in particolare modo sulla preferenza di una maggior chiarezza e facilità di pianificazione delle attività rispetto ad altri modelli.

Nella tabella 2.2, è riportata la pianificazione generale del progetto. In particolare viene presentato come si intende istanziare il modello di ciclo di vita a cascata in riferimento allo standard ISO/IEC 12207:1995.

Il totale della durata di 300 ore di stage, comprensive di attività lavorativa e di stesura di documentazione sono in accordo con quanto stabilito nel Dipartimento di Matematica Pura e Applicata dell'Università degli Studi di Padova.

I documenti da realizzarsi sono sottolineati e saranno ceduti all'azienda ARPAV al termine dello stage insieme al prodotto software *SEDAMS*.

Le ore preventive sono quelle dall'inizio del progetto.

<b>N°</b>	<b>Fasi ed attività del processo primario di sviluppo (con riferimento a ISO/IEC 12207:1995<sup>13</sup>)</b>	<b>Preventivo ore</b>	<b>Consuntivo ore</b>
1	<b>Analisi</b>	100	86
	Analisi dei requisiti di sistema (5.3.2).		
	Analisi dei requisiti software (5.3.4).		
2	<b>Progettazione</b>	89	96
	Progettazione architetturale del software (5.3.5).		
	Progettazione di dettaglio del software (5.3.6).		
3	<b>Realizzazione</b>	111	119
	Codifica software (5.3.7).		
	Integrazione software (5.3.8).		
	Collaudo software (5.3.9).		
	Integrazione sistema (5.3.10).		
	Collaudo sistema (5.3.11).		
<b>Totale</b>		<b>300</b>	<b>301</b>

Tabella 2.2. Istanziamento del modello di ciclo di vita a cascata con lo standard ISO/IEC 12207.

## 2.2. Dettaglio delle fasi del ciclo di vita

Questo paragrafo illustra in forma tabulare le attività previste per ogni fase, la documentazione da realizzarsi, i preventivi ed i consuntivi delle ore di stage.

### 2.2.1. Analisi

Si prevede la revisione di questa fase in data 28/04/06.

La tabella 2.3 descrive le attività previste per la fase di Analisi.

<sup>13</sup> La numerazione 5.3 è il processo di sviluppo nello standard ISO/IEC 12207:1995.

<b>Fase di Analisi</b>	<b>Preventivo ore</b>	<b>Consuntivo ore</b>
1. Analisi del problema. Studio dell'architettura del database.	14	10
2. Studio sulla reperibilità dei dati. Analisi delle eventuali tabelle da aggiungere nel database.	17	15
3. Stesura del documento <u>Piano di progetto</u> .	15	13
4. Stesura del documento <u>Analisi dei requisiti</u> .	22	18
5. Stesura del documento <u>Piano di qualifica v.1</u>	12	10
6. Inizio stesura del documento <u>Glossario</u> .	1	1
7. Studio del linguaggio PHP.	14	14
8. Verifica	5	5
<b>Totale</b>	<b>100</b>	<b>86</b>

Tabella 2.3. Fase di analisi.

Tutta la documentazione e le attività previste per questa fase sono state realizzate correttamente. Occorre tuttavia notificare una diminuzione di ore rispetto al preventivo pari a 33 ore. Seppure questo sia un errore di pianificazione significativo, non comporta un danno per lo svolgimento del progetto, dal momento che le ore preventive sono state sovrastimate. Tali 33 ore, saranno distribuite nelle due prossime fasi.

Di seguito, saranno descritte i possibili miglioramenti per ogni attività di questa fase, in modo tale da consentire in futuro di rendere più efficiente il processo di sviluppo.

1. Le ore effettive sono state in numero inferiore rispetto a quelle previste in quanto si era ritenuta più complessa l'analisi del problema e la parte di database da studiare. Allo stagista inoltre, è stato illustrato un esempio di file di esportazione in riferimento alla sezione che si occupa della Balneazione.
2. Da parte dell'azienda ARPAV, lo stagista ebbe a disposizione la documentazione di una precedente e parziale analisi della reperibilità dei dati sugli Alimenti. Circa un quarto delle informazioni sono state tratte da

questo documento.

3. Nella volontà di realizzare un Piano di progetto efficiente e di buona qualità, lo stagista ha utilizzato diversa documentazione. Per migliorare ulteriormente, occorre maggiore esperienza, che consiste anche nell'osservazione e nello studio di altri piani di progetto di diversi settori. La pianificazione ha riguardato il solo stagista coinvolto Dalle Pezze Piero. In questo senso non risulta molto complessa.
4. La realizzazione del documento di Analisi dei requisiti non ha incontrato problematiche, dal momento che i requisiti sono stati velocemente identificati. Si ritiene che un numero cospicuo di ore sarà necessariamente dedicato alla codifica, rispetto all'analisi ed alla progettazione.
5. Anche per questo documento, non si sono riscontrate problematiche. Per migliorare lo sviluppo del documento, vale la stessa considerazione specificata nel punto 3.
6. Per questo documento non si sono riscontrate problematiche.
7. Lo studio del linguaggio PHP, non ha riscontrato problematiche. Lo stagista ha avuto a disposizione dall'azienda ARPAV la documentazione Corso di PHP di Luca Balzerani. Lo stagista ha inoltre preso in considerazione la guida ufficiale del linguaggio PHP reperibile nel sito [www.php.net](http://www.php.net).
8. I dettagli sul resoconto della verifica effettuata sono reperibili nel documento Piano di qualifica. Senza alcun dubbio, un considerevole miglioramento a questo processo è quello in cui si prevede che la verifica sia effettuata da una componente diversa da quella che ha prodotto. Tuttavia, il progetto è stato concepito ed affidato al solo stagista Dalle Pezze Piero.

### **2.2.2. Progettazione**

Si prevede la revisione di questa fase in data 25/05/06.

La tabella 2.4 descrive le attività previste per la fase di Progettazione.

<b>Fase di Progettazione</b>	<b>Preventivo ore</b>		<b>Consuntivo ore</b>
	<b>Iniziale</b>	<b>Dopo 1 fase</b>	
1. Progettazione del software.	24	25	20
2. Stesura del documento <u>Specifica Tecnica</u> .	25	25	24
3. Stesura del documento <u>Piano di qualifica v. 2.</u>	8	8	9
4. Stesura del documento <u>Definizione di prodotto</u> .	14	14	20
5. Stesura del documento <u>Piano di qualifica v. 3.</u>	8	8	8
6. Verifica	10	13	15
<b>Totale</b>	<b>89</b>	<b>93</b>	<b>96</b>

Tabella 2.4. Fase di progettazione.

Tutta la documentazione e le attività previste per questa fase sono state realizzate correttamente. Rispetto alla preventivo, dopo la prima fase, si riscontra un aumento di 3 ore lavorative, dovuto principalmente ad un aumento di ore nella realizzazione del documento Definizione di prodotto. Questa lieve alterazione non comporta danni per lo svolgimento della fase successiva.

Di seguito, saranno descritti i possibili miglioramenti per ogni attività di questa fase, in modo tale da consentire in futuro di rendere più efficiente il processo di sviluppo.

1. Il documento di Analisi dei requisiti presentava i requisiti di sistema e software in modo esaustivo. Inoltre, la realizzazione del documento Codifica esportazione, ha permesso di ottenere una chiara visione di come i campi devono essere esportati. Al termine della prima fase, sussistevano otto campi per i quali non si conosceva la locazione all'interno del database. Tali campi al termine della seconda fase sono stati ridotti a due, mentre gli altri, sono stati individuati. Per questi motivi, la progettazione è stata fatta con maggior velocità, ossia perché era già molto chiaro cosa il software necessitava.
2. Il documento Specifica tecnica è stato realizzato correttamente utilizzando

diversi diagrammi UML. L'unico miglioramento consiste in una maggiore esperienza nell'utilizzo di UML.

3. La seconda versione del documento Piano di qualifica, non ha riscontrato errori nel suo svolgimento, eccetto il fatto che si è preferito, per motivi di ridondanza, porre la pianificazione e l'esito delle verifiche nella medesima tabella. (Vedere capitolo 4 del documento Piano di qualifica).
4. Il documento Definizione di prodotto, non ha determinato difficoltà cruciali, durante la sua realizzazione. Per ottimizzare ulteriormente il suo svolgimento, potrebbe essere stato utile, se esiste, l'utilizzo di software per la modellazione UML con generazione automatica di codice PHP.
5. La terza versione del documento Piano di qualifica, è stata realizzata correttamente senza sollevare nessuna attività anomala.
6. La verifica di questa fase è avanzata correttamente e documentata come pianificato nel paragrafo 4.1, del documento Piano di qualifica.

### **2.2.3. Realizzazione**

Si prevede la revisione di questa fase in data 10/06/06.

La tabella 2.5 descrive le attività previste per la fase di Realizzazione.



<b>Fase di Realizzazione</b>	<b>Preventivo ore</b>			<b>Consuntivo ore</b>
	<b>Iniziale</b>	<b>Dopo 1 fase</b>	<b>Dopo 2 fase</b>	
1. Codifica.	40	50	57	56
2. Stesura del documento <u>Manuale utente v.1.</u>	8	8	8	8
3. Stesura del documento <u>Piano di qualifica v.4</u>	10	10	8	8
4. Test di unità.	15	15	11	12
5. Test di integrazione.	10	10	8	8
6. Stesura del documento <u>Piano di qualifica v.5.</u>	5	5	5	5
7. Stesura del documento <u>Manuale utente v.2.</u>	7	7	7	8
8. Test di sistema.	8	8	7	7
9. Collaudo col sistema.	8	8	7	7
<b>Totale</b>	<b>111</b>	<b>121</b>	<b>118</b>	<b>119</b>

Tabella 2.5. Fase di realizzazione.

Tutta la documentazione pianificata per questa fase è stata prodotta. Rispetto alla pianificazione delle ore prevista dopo la seconda fase, la differenza è minimale (1 ora). Di seguito è riportato il resoconto delle attività svolte.

1. La codifica del software *SEDAMS* è stata realizzata nella precedente fase, e completata in questa. Non sono state rilevate difficoltà o errori rilevanti come può testimoniare il documento Piano di qualifica. Parte del successo di questa attività è dovuta alla somiglianza dei linguaggi di programmazione PHP e C ed alla documentazione in rete ricca di esempi disponibili.
2. La prima versione del manuale utente non ha subito né rallentamenti né anticipazioni. Il documento non è stato particolarmente impegnativo in quanto non ci sono molte funzionalità nel software *SEDAMS*.
3. Il piano di qualifica è stato sviluppato coerentemente con la sua pianificazione.
4. I test di unità hanno rilevato errori soprattutto di distrazione, ma non

concettuali. Dopo la loro correzione, non sono stati rilevati altri errori né imprecisioni.

5. I test di integrazione sono stati svolti coerentemente alla pianificazione prevista. La durata dei test di integrazione è dovuta essenzialmente al rilevante tempo di esecuzione del software nel prelevamento dei dati da un'altra provincia, e nella scrittura su file.
6. La quinta ed ultima versione del piano di qualifica è stata svolta con profitto seguendo la pianificazione prevista. Non sono stati rilevati problemi.
7. Come seconda versione del manuale utente, si è prevista la scrittura della guida in linea per l'utente, richiamabile facilmente dal software. Nella realizzazione della pagina web, si è eseguita un'esportazione del documento Manuale utente in formato HTML ed una successiva “pulizia” del codice, in modo tale che risulti facile ed immediato modificare o integrare il documento.
8. I test di sistema eseguiti non hanno rilevato problemi. La durata prevista è stata quella effettiva ed è dovuta ai tempi di esportazione come specificato nel punto 5.
9. Il collaudo con il sistema è avvenuto con profitto e anche per questo, i tempi sono i medesimi di quelli dei punti 7 e 5.

### ***3. Gestione dei rischi di progetto***

Questo capitolo descrive la modalità di gestione dei rischi che questo piano di progetto può comportare:

#### ***3.1. Piano di progetto inefficace***

La possibilità che questo piano di progetto si dimostri inefficace rispetto a quello che si è previsto nella sua formulazione, costituisce il rischio maggiore. Per tale motivo, su scadenza settimanale, questo documento sarà supervisionato dal tutor aziendale Zambotto Paolo, in modo tale da poter prevenire eventuali futuri discostamenti organizzativi.

#### ***3.2. Gestione del piano di progetto inadeguata***

Il rischio di un' inadeguata gestione del piano di progetto è stata valutata e se ne propone una soluzione nel [capitolo 5](#).

#### ***3.3. Difficoltà di apprendimento del linguaggio PHP***

Nella Laurea Triennale di Informatica dell'Università degli Studi di Padova, non è previsto in alcun corso, lo studio del linguaggio PHP. Come principale documentazione di riferimento sarà utilizzata la documentazione ufficiale reperibile nel sito <http://www.php.net/>. Nel caso in cui si constataste la possibilità di un ritardo di apprendimento del linguaggio rispetto al tempo preventivato, lo stagista contatterà il tutor aziendale Zambotto Paolo, per chiarire in modo immediato tali difficoltà riscontrate. La richiesta di informazioni riguarderà soltanto questioni prettamente riguardanti aspetti del PHP e non nozioni sui linguaggi di programmazione appresi durante il corso di laurea.

## ***4. Gestione della qualità del piano di progetto***

Per garantire un'apprezzabile qualità del Piano di progetto, si prevede che ogni attività venga svolta secondo il ciclo *PDCA* e soprattutto venga documentata. Ogni documento quindi fornirà un registro delle modifiche, semplice ma intuitivo, in modo tale da registrare ogni cambiamento effettuato. Ad ogni modifica inoltre corrisponderà una versione di documento. In tal modo, lo stagista potrà verificare immediatamente l'introduzione di errori in ogni processo software svolto, garantendo migliore qualità ed efficienza.

I dettagli invece su come si intende gestire la qualità del progetto stesso e le procedure di controllo per garantire la qualità di processo, sono presenti nel documento Piano di qualifica.

Il ciclo *PDCA* per il Piano di progetto è così costituito:

- **Pianificazione:**

Si pianificano i processi, le attività da svolgersi e le scadenze temporali.

- **Esecuzione:**

Si attende che le attività e i processi avanzino (come pianificati).

- **Controllo:**

Si controlla che la pianificazione sia stata rispettata. In caso contrario, si focalizza il problema e lo si documenta.

- **Azione:**

Si pianificano i processi e le attività futuri in base all'esito del controllo. Nel caso il controllo restituisse un valore differente da quello atteso, si documenta come si sarebbe potuto evitare il problema insorto (miglioramento del processo).

## ***5. Gestione del piano di progetto***

Tale piano di progetto è gestito dallo stagista Dalle Pezze Piero considerando lo sviluppo del progetto *SEDAMS* ed i colloqui tra lo stagista stesso e il tutor aziendale Zambotto Paolo. Le dichiarazioni ottenute durante questi colloqui serviranno per migliorare la pianificazione prevenendo eventuali problematiche discusse nel [capitolo 3](#).

Durante questi incontri, lo stagista fornirà i documenti al tutor aziendale, illustrando l'avanzamento delle attività e una stima dei discostamenti.

Al termine di ogni fase, come documentato nel [paragrafo 2.1](#), lo stagista contatterà il docente universitario che rappresenta il suo tutor universitario per discutere lo stato di avanzamento ed eventuali difficoltà riscontrate o variazioni di pianificazione.

## **Piano di qualifica**

**Nome del prodotto:**

Sistema di Esportazione Dati Alimenti per il Ministero della Salute  
(SEDAMS).

**Data:**

10/06/06

**Versione:**

5.0

**Stato del documento:**

Formale esterno

**Redazione:**

Piero Dalle Pezze

**Revisione:**

Piero Dalle Pezze    Padova, li 10/06/06    Firma .....

**Approvazione:**

Paolo Zambotto    Padova, li 10/06/06    Firma .....

**Lista di distribuzione:**

1. Paolo Zambotto
2. Piero Dalle Pezze

### **Registro delle modifiche:**

- v 5.0 08/06/06  
Revisione formale del documento.
- v 4.3 06/06/06  
Verifica della documentazione.
- v 4.2 05/06/06  
Completamento test di collaudo.
- v 4.1 02/06/06  
Verifica della documentazione del manuale utente.
- v 4.0 01/06/06  
Esecuzione dei test di integrazione. Inizio test di collaudo.
- v 3.1 25/05/06  
Verifica della documentazione.
- v 3.0 24/05/06  
Pianificazione test di integrazione e di sistema.
- v 2.2 23/05/06  
Riempita tabella 4.17.
- v 2.1 22/05/06  
Riempita tabella 4.16.
- v 2.0 19/05/06  
Inserita e riempita tabella 4.15 nel paragrafo 4.3. Riempite tabelle 4.6 e 4.7 nel paragrafo 4.2.
- v 1.2 14/05/06  
Pianificazione test di unità.
- v 1.1 04/05/06  
Inserimento delle tabelle nel capitolo 4.1.
- v 1.0 28/04/06  
Revisione formale del documento.
- V 0.3 21/04/06  
Inserito paragrafo 4.1. Redatta tabella 4.1

v 0.2 20/04/06

Inserito capitolo 2, 3, 4, 5.

v 0.1 19/04/06

Creazione del documento.



## Sommario:

Il documento presenta la strategia e la documentazione di verifica e validazione proposta dallo stagista al fine di garantire la qualità del prodotto software *SEDAMS*.

## Indice:

1. Introduzione	74
1.1. Scopo del documento	74
1.2. Scopo del prodotto	74
1.3. Glossario	74
2. Visione generale della strategia di verifica	75
2.1. Organizzazione, pianificazione strategica e temporale	75
2.2. Risorse necessarie, risorse disponibili	76
2.3. Strumenti, tecniche, metodi	76
3. Gestione amministrativa della revisione	78
3.1. Comunicazione risoluzione di anomalie	78
3.2. Trattamento delle discrepanze	79
3.3. Procedure di controllo di qualità di processo	79
4. Resoconto delle attività di verifica	80
4.1. Verifica sulla documentazione	80
4.2. Tracciamento componenti – requisiti	84
4.3. Dettaglio delle verifiche tramite analisi	85
4.4. Dettaglio delle verifiche tramite test	86
4.4.1. Test di unità	86
4.4.2. Test di integrazione	89
4.4.3. Test di sistema	90
5. Pianificazione ed esecuzione del collaudo	91
5.1. Pianificazione ed esito della campagna di validazione	91

# ***1. Introduzione***

## ***1.1. Scopo del documento***

Lo scopo di questo documento è di illustrare come lo stagista Dalle Pezze Piero intende pianificare ed attuare la campagna di verifica e validazione nell'analisi, progettazione e realizzazione del progetto *SEDAMS*.

## ***1.2. Scopo del prodotto***

Lo scopo del prodotto è quello di esportare dati nella codifica richiesta dal Ministero della Salute specificata nel documento *SA-99-D1006-015*.

## ***1.3. Glossario***

Il glossario viene fornito come documento allegato ed è unico per tutta la documentazione.

## ***2. Visione generale della strategia di verifica***

### ***2.1. Organizzazione, pianificazione strategica e temporale, responsabilità***

Dal momento che il progetto *SEDAMS* è svolto unicamente dallo stagista, sorge un conflitto di interesse dovuto al fatto che il realizzatore ed il verificatore sono la medesima persona. Si ritiene per cui che la documentazione ed il colloquio frequente con i tutor siano ancora più necessari proprio per risolvere in parte a questo conflitto.

Lo stagista dall'altra parte adopererà il ciclo *PDCA* al prodotto nel seguente modo:

- **Pianificazione:**

Sarà determinata principalmente dai documenti Piano di progetto e Piano di qualifica. Il piano di progetto conterrà tutte le informazioni indispensabili, per la gestione lavorativa dello stagista. Saranno per cui documentate le attività da svolgersi definendo cosa e come ognuna di esse deve essere compiuta. Il piano di qualifica, invece, definirà la metodologia di verifica di ogni processo.

- **Esecuzione:**

Comporta l'esecuzione delle attività previste nel piano di progetto per il processo software coinvolto. Per esempio ricerca ed analisi dei requisiti e stesura del documento.

- **Controllo:**

Nel caso di verifica, si utilizza il piano di qualifica ed il prodotto di quel specifico processo controllando se i requisiti pianificati sono quelli ottenuti.

Nel caso di validazione, si utilizzano i documenti realizzati precedentemente controllando il soddisfacimento di tutti i requisiti.

I controlli effettuati devono essere quelli pianificati. I risultati di ogni

controllo sarà documentato nel piano di qualifica e testimonierà l'uscita da quel processo o da quella fase.

- **Azione:**

A seconda che il controllo riveli la presenza di *fallimento, guasto o errore*, sarà necessario intervenire nel documento realizzato e ripetere il ciclo *PDCA*.

Ogni settimana, lo stagista arresterà l'avanzamento del progetto assumendo il ruolo di verificatore, impegnandosi a controllare l'evoluzione del processo e del prodotto.

Per quanto concerne la documentazione, in generale, sarà adoperata la tecnica *walkthrough*, leggendo quindi i documenti e rilevando errori logici, grammaticali, formali e concettuali.

I dettagli della pianificazione e dell'esito della campagna di verifica sono presenti nel [capitolo 4](#) di questo documento.

## ***2.2. Risorse necessarie, risorse disponibili***

Le risorse software impiegate sono:

- OpenOffice 2.0
- Oracle 8.1.6
- Intel Pentium IV 1700 MHz, 130 MB di RAM
- Sistema operativo Microsoft Windows 2000 Professional
- Notebook personale.
- Apache 2.0.49
- XEmacs 21.4.19 e GNU Emacs
- Browser Mozilla Firefox 1.5.0.4
- Linguaggio PHP versione 5.1.2

## ***2.3. Strumenti, tecniche, metodi***

Si utilizzeranno le seguenti tecniche e metodi di verifica:

- Tracciamento, per verificare la correttezza e l'economicità dell'implementazione in fase di progettazione.
- Analisi statica:
  - *Inspection*: metodo che prevede una lettura del codice focalizzata solo sulle zone più a rischio di errori. Questo tipo di analisi statica verrà utilizzata essenzialmente solo sul codice riguardante l'interfaccia grafica.
  - *Walkthrough*: metodo di lettura del codice che verrà applicato durante la verifica della fase realizzativa. Questa attività, che sarà pianificata, avrà l'obiettivo di individuare i difetti del codice, percorrendolo e simulandone possibili esecuzioni.
  - Analisi di flusso di controllo: accerterà che il codice venga eseguito in modo corretto. Questo implica un codice ben strutturato e la localizzazione di parti di codice non raggiungibili.
  - Analisi di flusso dei dati: accerterà che non si utilizzino variabili non inizializzate o comunque prive di valore in modo da rilevare possibili anomalie.
- I test dinamici saranno sia di tipo *black box*, che di tipo *white box*.

### ***3. Gestione amministrativa della revisione***

#### ***3.1. Comunicazione e risoluzione di anomalie***

Eventuali anomalie riscontrate prima della consegna del prodotto al tutor aziendale Zambotto Paolo, sono sotto la responsabilità dello stagista Dalle Pezze Piero, il quale si impegna a risolverle.

Le comunicazioni di anomalie possono pervenire da parte del tutor aziendale durante i colloqui o utilizzando una delle caselle di posta elettronica qui indicate:

- [pdallepezze@arpa.veneto.it](mailto:pdallepezze@arpa.veneto.it)
- [pdallep@studenti.math.unipd.it](mailto:pdallep@studenti.math.unipd.it)

Le comunicazioni di anomalie devono contenere le seguenti informazioni:

- Nome e Cognome mittente.
- Data riscontro dell'anomalia.
- Natura dell'anomalia.
- Descrizione dello scenario e delle condizioni sotto le quali si è verificata l'anomalia.

Alla ricezione di una comunicazione di anomalia, lo stagista valuterà opportunamente la gravità dell'accaduto ed effettuerà una stima del danno e del lavoro per il rimedio. Dopo di che deciderà se contattare nuovamente il tutor nel caso l'anomalia non fosse dovuta al software *SEDAMS* o fosse di grave entità. Nel caso in cui il rimedio fosse ridotto, lo stagista si occuperà direttamente della sua risoluzione ed in seguito contatterà il tutor aziendale delle modifiche apportate.

### ***3.2. Trattamento delle discrepanze***

Qualora la procedura di trattamento delle discrepanze (dallo studio alla attuazione della soluzione) risultasse inaccettabile in termini di tempo e costi per il cliente o si presentasse come una situazione di elevato rischio di fallimento del progetto, la discrepanza non verrà risolta o verrà risolta solo in parte (secondo accordi che vengono presi col cliente in questa situazione).

Altrimenti, eventuali discrepanze riscontrate prima della consegna del prodotto al tutor aziendale Zambotto Paolo, sono sotto la responsabilità dello stagista Dalle Pezze Piero, il quale si impegna a risolverle. Per prevenire a queste, saranno frequenti i colloqui tra lo stagista ed il suo tutor aziendale.

### ***3.3. Procedure di controllo di qualità di processo***

Per garantire la qualità di processo, si ritiene necessario documentare anche l'evoluzione del processo stesso, in particolar modo riferendone la qualità del suo sviluppo. Si cercherà quindi di applicare il ciclo *PDCA* ai processi stessi come specificato di seguito:

- **Pianificazione:**  
Si pianificherà come il processo dovrà svolgersi.
- **Esecuzione:**  
Consisterà nella documentazione di come il processo è stato svolto.
- **Controllo:**  
Si controllerà l'esito del processo con quanto atteso nella pianificazione del processo stesso.
- **Azione:**  
Si documenterà come il processo può essere migliorato, in riferimento al controllo.

La pianificazione, l'esecuzione, la registrazione del controllo e l'azione saranno documentati nel [capitolo 2.2](#) del Piano di progetto.

## 4. Resoconto delle attività di verifica

### 4.1. Verifica sulla documentazione

Le tabelle che seguono descrivono la pianificazione e l'esito delle verifiche sui documenti prodotti.

Le tabelle 4.1, 4.2, 4.3, 4.4 illustrano l'esito della verifica dei documenti previsti nella fase di analisi.

<b>Nome del documento:</b> <u>Analisi dei requisiti</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Conflittualità tra requisiti (funzionali e non funzionali).</b>	Nessun conflitto rilevato.
<b>Requisiti mal specificati.</b>	1. Il requisito RFC1 della versione 0.3 è stato suddiviso in RFC1 e RFC2.
<b>Errori nei diagrammi presentati.</b>	1. La figura 3.1 della versione 0.3 è stata riformulata, per un errore concettuale di funzionamento.
<b>Conflitti e/o mancanze tra diagrammi (anche use case narrativi) e requisiti funzionali.</b>	Nessun conflitto rilevato.

Tabella 4.1. Verifica del documento Analisi dei requisiti.

<b>Nome del documento:</b> <u>Piano di qualifica v1</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di incompletezze nel ciclo PDCA.</b>	Non si registrano incompletezze.
<b>Ricerca di verifiche non eseguite.</b>	Nessuna.

Tabella 4.2. Verifica del documento Piano di qualifica v1.



<b>Nome del documento:</b> <u>Piano di progetto</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di incompletezze nella pianificazione delle attività.</b>	Non ci sono attività incomplete. Le ore effettive sono inferiori alle ore previste (33 ore in meno). Tali ore andranno suddivise nelle due fasi successive.
<b>Ricerca di rischi incontrati e non gestiti.</b>	Non sono stati rilevati errori ne notificati rischi.

Tabella 4.3. Verifica del documento Piano di progetto.

<b>Nome del documento:</b> <u>Glossario</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di termini tecnici o sigle nei documenti che non compaiono nel Glossario.</b>	Non ci sono termini tecnici utilizzati che non compaiono nel glossario.

Tabella 4.4. Verifica del documento Glossario.

Le tabelle 4.5, 4.6, 4.7, 4.8, 4.9 illustrano l'esito della verifica dei documenti previsti nella fase di progettazione.

<b>Nome del documento:</b> <u>Specifica tecnica</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di inconsistenze con il documento <u>Analisi dei requisiti</u>.</b>	Non sono state riscontrate inconsistenze con il documento <u>Analisi dei requisiti</u> .
<b>Ricerca di mancanze rispetto al documento <u>Analisi dei requisiti</u>.</b>	Non sono state rilevate mancanze. La matrice di tracciamento non presenta lacune.
<b>Ricerca di errori nei diagrammi presentati.</b>	I diagrammi non presentano errori. Nel corso della revisione è stato anche ascoltato il tutor aziendale.

Tabella 4.5. Verifica del documento Specifica tecnica.

<b>Nome del documento:</b> Definizione di prodotto	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di inconsistenze con i documenti <u>Analisi dei requisiti</u> e <u>Specificazione tecnica</u>.</b>	Non sono state riscontrate inconsistenze con il documento <u>Analisi dei requisiti</u> .
<b>Ricerca di mancanze rispetto al documento <u>Analisi dei requisiti</u> e <u>Specificazione tecnica</u>.</b>	Non sono state rilevate mancanze rispetto ai documenti <u>Analisi dei requisiti</u> e <u>Specificazione tecnica</u> .
<b>Ricerca di errori nei diagrammi presentati.</b>	Non sono stati rilevati errori nei diagrammi presentati, inconsistenze con quelli presentati nel documento <u>Specificazione tecnica</u> e con il codice sorgente.

Tabella 4.6. Verifica del documento Definizione di prodotto.

<b>Nome del documento:</b> Piano di qualifica v.2	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di incompletezze nel ciclo PDCA.</b>	Non si sono riscontrati errori nell'esecuzione del ciclo PDCA.
<b>Ricerca di verifiche non eseguite.</b>	Non si sono riscontrate verifiche non eseguite.

Tabella 4.7. Verifica del documento Piano di qualifica v.2.

<b>Nome del documento:</b> Piano di qualifica v.3	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di incompletezze nel ciclo PDCA.</b>	Non si sono riscontrati errori nell'esecuzione del ciclo PDCA.
<b>Ricerca di verifiche non eseguite.</b>	Non si sono riscontrate verifiche non eseguite.

Tabella 4.8. Verifica del documento Piano di qualifica v.3.

<b>Nome del documento:</b> <u>Glossario</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di termini tecnici o sigle nei documenti che non compaiono nel Glossario.</b>	Non sono stati aggiunti termini al glossario.

Tabella 4.9. Verifica del documento Glossario.

Le tabelle 4.10, 4.11, 4.12, 4.13, 4.14 illustrano l'esito della verifica dei documenti previsti nella fase di realizzazione.

<b>Nome del documento:</b> <u>Manuale utente v.1</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di mancanze nella descrizione del funzionamento del prodotto.</b>	Nessuna mancanza nella descrizione del funzionamento del prodotto rilevata.
<b>Ricerca di messaggi di errore non descritti.</b>	Nessun messaggio di errore mancante.
<b>Ricerca di parti non chiare o poco esplicative per l'utente.</b>	Non sono state rilevate parti poco chiare o non esplicative. Il test è stato sostenuto anche da utenti non del mestiere con il medesimo esito.
<b>Ricerca di termini non presenti nel glossario, ma di difficile comprensione.</b>	Nessun termine tecnico non presente nel glossario.

Tabella 4.10. Verifica del documento Manuale utente v.1.

<b>Nome del documento:</b> <u>Manuale utente v.2</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di inconsistenze tra manuale utente e guida in linea.</b>	Nessuna inconsistenza rilevata.

Tabella 4.11. Verifica del documento Manuale utente v.2.

<b>Nome del documento:</b> <u>Piano di qualifica v.4</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di incompletezze nel ciclo PDCA.</b>	Nessuna incompletezza è stata rilevata durante il ciclo <i>PDCA</i> .
<b>Ricerca di verifiche non eseguite.</b>	Nessuna verifica è risultata non eseguita totalmente o parzialmente.

Tabella 4.12. Verifica del documento Piano di qualifica v.4.

<b>Nome del documento:</b> <u>Piano di qualifica v.5</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di incompletezze nel ciclo PDCA.</b>	Nessuna incompletezza è stata rilevata durante il ciclo <i>PDCA</i> .
<b>Ricerca di verifiche non eseguite.</b>	Non sono state rilevate verifiche pianificate, ma non eseguite parzialmente o totalmente.

Tabella 4.13. Verifica del documento Piano di qualifica v.5.

<b>Nome del documento:</b> <u>Glossario</u>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di termini tecnici o sigle nei documenti che non compaiono nel Glossario.</b>	Mancavano i termini tecnici <i>form</i> , <i>frame</i> . Tali termini sono stati inseriti.

Tabella 4.14. Verifica del documento Glossario.

## 4.2. Tracciamento componenti – requisiti

Per il tracciamento delle componenti con i requisiti, si rimanda al capitolo 5 dei documenti Specificazione tecnica e Definizione di prodotto.

### 4.3. Dettaglio delle verifiche tramite analisi

Nella tabella 4.15, è descritta la pianificazione e l'esito delle analisi statiche attuate nel codice sorgente.

<b>Analisi statica del codice sorgente</b>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca errori nella formulazione della select.</b>	Nessun errore riscontrato nelle select di prelevamento analisi e campioni.
<b>Ricerca di campi di default che non vengono inizializzati o non presenti.</b>	Nessun campo dato non inizializzato o non presente.
<b>Ricerca di codice non conforme alla codifica stabilita nel paragrafo 2.4 del documento <u>Definizione di prodotto</u>.</b>	Non sono stati ritrovati parti di codice non conformi.
<b>Ricerca di parti di codice non raggiungibili.</b>	Non sono state rilevate parti di codice non raggiungibili.
<b>Ricerca di parti di codice complesse che possono essere semplificate.</b>	La formattazione e la concatenazione dei campi da esportare (metodi esporta__()) sono stati separati dal metodo esegui(\$filename, \$filename_err) della classe EsportatoreAlimenti.
<b>Ricerca di conversioni di tipo mal effettuate.</b>	Non sono stati riscontrati errori dovuti a conversioni di tipo.
<b>Ricerca di stringhe dei campi dei campioni e delle analisi non corrette.</b>	Ritrovati i seguenti errori nella formattazione: <ol style="list-style-type: none"><li>1. Il campo prelevato \$numerical_entry (metodo esporta(\$filename, \$filename_err)), tralasciava lo zero davanti al punto (es. .342). Il campo è stato corretto (0,342).</li><li>2. Il campo \$segno non è esportato. In particolare, set_segno(\$entry_qualifier) non ritornava la stringa da esportare.</li></ol> Dopo la correzione nessun errore rilevato.
<b>Ricerca di stringhe di lunghezze scorrette nel record di continuazione.</b>	Non sono stati ritrovati errori.
<b>Ricerca di record in cui non ci sono 16 analisi (record di tipo 01) o 31 (record di continuazione). In particolare ricercare errori nella lunghezza della stringa di completamento.</b>	Ritrovati i seguenti errori. <ol style="list-style-type: none"><li>1. Il metodo esporta(\$filename, \$filename_err) avrebbe stampato 17 analisi per il record di tipo 01 e 32 analisi per il record di tipo continuazione.</li></ol> Dopo la correzione, nessun errore rilevato.
<b>Ricerca di documentazione del codice non conforme a quanto stabilito nel</b>	Non sono stati ritrovati errori.

<b>Analisi statica del codice sorgente</b>	
<b>paragrafo 2.2 del documento <u>Definizione di prodotto</u>.</b>	
<b>Ricerca di eccezioni nell'inserimento dei parametri che non sono gestite.</b>	Non sono state ritrovate eccezioni non gestite.
<b>Ricerca di eccezioni che possono essere sollevate nel corso dell'esportazione dei dati, che non sono gestite.</b>	Non sono state ritrovate dimenticanze.

Tabella 4.15. Resoconto dell'analisi statica.

#### **4.4. Dettaglio delle verifiche tramite test**

Di seguito sono riportate la pianificazione e l'esito delle verifiche dinamiche sul software *SEDAMS*. In questo capitolo si forniscono i dettagli dei test di unità, di integrazione e di sistema. Nel [capitolo 5](#), invece, si presentano i dettagli dei test di collaudo.

##### **4.4.1. Test di unità**

Nelle seguenti tabelle si riportano la pianificazione e l'esito dei test di unità effettuati. Si considerano le due unità:

1. Esportatore. Comprende i moduli Esportatore e EsportatoreAlimenti. (Tabella 4.16).
2. GUI. La GUI è stata progettata e realizzata nei linguaggi PHP ed HTML senza l'ausilio di classi ed interfacce. L'unità GUI comprende quindi tutti gli altri file che realizzano l'interfaccia grafica utente. (Tabella 4.17).

<b>Unità: Esportatore</b>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Verifica sul prelevamento: effettuare il prelevamento dei campi richiesti dei soli campioni e la loro stampa. Ricercare, mediante il software TOAD, dati prelevati scorretti.</b>	Nessun errore rilevato.
<b>Verifica sul prelevamento: per ogni campione prelevato, effettuare il prelevamento delle sue analisi e la loro stampa. Ricercare, mediante il software TOAD, dati prelevati scorretti.</b>	Nessun errore rilevato.
<b>Verifica sull'esportazione: nel path corrente, effettuare l'esportazione di campioni ed analisi di un certo periodo. Ricercare errori nell'esportazione dei record di tipo "01". Verificare la lunghezza dei record "01" diverse da 1000 e dei campi diverse da quelle prescritte nel documento SA-99-D1006-015.</b>	Rilevati i seguenti errori: <ol style="list-style-type: none"> <li>1. non viene esportato il campo \$lab_sec_istanza;</li> <li>2. non viene esportato il campo \$esito_globale.</li> </ol> Dopo la correzione, nessun errore rilevato. In allegato si consegna anche il file di esportazione exp_01.txt, che illustra un'esportazione regolare di record di tipo "01".
<b>Come sopra, ma per i record di continuazione.</b>	Rilevati i seguenti errori: <ol style="list-style-type: none"> <li>1. non viene esportato correttamente il completamento del record fino al 1000 byte. In particolare, sussiste un errore nel test che stabilisce la lunghezza del completamento. (Se completamento per record di tipo "01" o di tipo continuazione).</li> </ol> Dopo la correzione, nessun errore rilevato. (Neppure nei punti precedenti).
<b>Come sopra, ma prelevare campioni con numero analisi appartenente ai seguenti insiemi: {0}, {1,...,15}, {16}, {17,...,?} . Per ottenere tale scopo, rimuovere, se necessario, qualche filtro nelle select.</b>	Nessun errore rilevato. In allegato si consegna il file exp_cont.txt, il quale illustra un'esportazione anche con record di continuazione. A causa di dati mancanti nel database, ma che dovranno in futuro essere riempiti, si è rimosso il filtro "COMPONENT.AV_CAMPO1 IS NOT NULL" nella select del prelevamento delle analisi. In tal modo è stato possibile prelevare ed esportare un maggior numero di analisi per campione.
<b>Verifica sul file di esportazione: Ricercare divergenze nelle lunghezze dei campi e dei dati con il file di esportazione, generato, mediante il</b>	Nessun errore rilevato. In allegato è consegnato il file di esportazione, generato, mediante il software consegnato dal Ministero della Salute Italiana, dalla sede ARPAV di Verona. Tale file è denominato exp_modello.txt

<b>Unità: Esportatore</b>	
<i>software consegnato dal Ministero della Salute Italiana, dalla sede ARPAV di Verona.</i>	
<i>Prelevamento ed esportazione di campioni ed analisi con numerosi campi nulli. Testare errori nella formattazione dei valori di default.</i>	Nessun errore rilevato.

Tabella 4.16. Pianificazione ed esito dei test per l'unità Esportatore.

<b>Unità: GUI</b>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<i>Ricerca di errori nei collegamenti ipertestuali realizzati.</i>	Nessun errore rilevato.
<i>Ricerca di parametri mancanti nella form.</i>	Nessun errore rilevato.
<i>Verifica delle date: verificare errori nell'inserimento del periodo di esportazione. In particolare verificare la non validità della data di inizio, di quella di fine e l'inconsistenza del periodo.</i>	Nessun errore rilevato.
<i>Verificare che l'utente non sia avvisato dopo l'inserimento di date scorrette.</i>	Nessun errore rilevato.
<i>Verificare che i campi dati della form vengono passati correttamente alla pagina verifica.php .</i>	Nessun errore rilevato.
<i>Verificare che il path di scrittura del file sia scorretto. In particolare verificare che i casi di creazione delle cartelle.</i>	Rilevati i seguenti errori: 1. Non viene creata la cartella del giorno di esportazione nella cartella export/. Dopo la correzione, nessun errore rilevato.
<i>Verificare che la durata della transazione non sia</i>	Nessun errore rilevato.



<b>Unità: GUI</b>	
<i>corretta. In particolare verificare la funzione <code>get_durata(\$tempo_inizio, \$tempo_fine)</code> nel file <code>sedams.php</code>.</i>	

Tabella 4.17. Pianificazione ed esito dei test per l'unità GUI.

#### 4.4.2. Test di integrazione

I seguenti test, riportati nella tabella 4.18, forniscono i test di integrazione tra le unità GUI ed Esportatore.

<b>Integrazione delle unità: Esportatore e GUI</b>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<i>Ricerca di dati della form che vengano passati scorrettamente al costruttore <code>EsportatoreAlimenti()</code>.</i>	Nessun errore rilevato.
<i>Ricerca di errori nella generazione delle statistiche.</i>	Nessun errore rilevato.
<i>Ricerca di errori nella visualizzazione dello stato della transazione.</i>	Nessun errore rilevato. Migliorata la visualizzazione, introducendo opportuni paragrafi.
<i>Ricerca di errori nell'inizializzazione del path e nella scrittura del file. In particolare ricercare errori nella creazione del file. (Deve essere dentro il path corretto).</i>	Nessun errore rilevato.
<i>Ricercare errori nei link ai file di esportazione generati. In particolare, testare l'esistenza di link scorretti o di files mancanti per quei links.</i>	Nessun errore rilevato.

Tabella 4.18. Pianificazione ed esito dei test di integrazione.

#### 4.4.3. Test di sistema

<b>Test di sistema</b>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Ricerca di anomalie nell'esecuzione normale del sistema. (Apertura, esecuzione, visualizzazione).</b>	Nel browser Microsoft Internet Explorer, possono sorgere alcuni errori di visualizzazione nell'apertura del file di esportazione (formato .dax). Aprendo con un comune editor di testo o un altro browser, non sono state rilevate anomalie nella visualizzazione.
<b>Ricerca di anomalie nell'utilizzo ripetuto del software. (Esportazione da diversi server DB_LIMS nella stessa sessione).</b>	Nessuna anomalia rilevata.
<b>Ricerca di anomalie nell'utilizzo ripetuto del software. (Esportazione ripetuta dallo stesso server DB_LIMS nella stessa sessione, prima con gli stessi parametri, poi con parametri diversi. Verificare anomalie nel file prodotto).</b>	Nessuna anomalia rilevata.

Tabella 4.19. Pianificazione ed esito dei test di sistema.

## 5. Pianificazione ed esecuzione del collaudo

### 5.1. Pianificazione ed esito della campagna di validazione

Nella tabella 5.1, si riporta la pianificazione e l'esito del test di collaudo che si intende eseguire.

<b>Test di collaudo</b>	
<b>Pianificazione di verifica</b>	<b>Esito</b>
<b>Effettuare un'esportazione regolare della durata di un anno. Il sistema deve monitorare lo stato della transazione ed esportare i dati nel file.</b>	Nessun errore rilevato. In allegato a questo documento è consegnato il file exp_1anno.txt.
<b>Immissione scorretta dei parametri (formato scorretto). Il sistema deve visualizzare i campi contenenti gli errori.</b>	Nessun errore rilevato.
<b>Immissione di date inesistenti (es. 29/02/2005). Il sistema deve visualizzare i campi contenenti gli errori.</b>	Nessun errore rilevato.
<b>Interrompere la connessione durante l'esportazione. Il sistema deve interrompere la transazione, segnalare la caduta di connessione e resettare il suo stato temporaneo.</b>	Nessun errore rilevato.
<b>Inserire il file esportato nel programma del Ministero della Salute, a disposizione dell'azienda ARPAV e testare che tutti i dati inseriti siano quelli attesi.</b>	Per motivi di mancata risposta entro il termine dello stage dello studente Dalle Pezze Piero, questo test non è stato eseguito. Tuttavia, si è provveduto a confrontare con un editor che permettesse di visualizzare il numero della colonna della riga, un file di esportazione prodotto dal software del Ministero e quello prodotto dal software SEDAMS, e non sono state rilevate divergenze né nelle lunghezze dei campi, né nei campi esportati.
<b>Effettuare un'esportazione più lunga della durata di 1 anno. Rilevare anomalie nel funzionamento del sistema a lungo termine.</b>	Sono state eseguite quattro esportazioni di periodi lunghi (3 anni e di 10 anni), dai server DBVRLIMS e DBPDLIMS. Non sono state rilevate anomalie nell'esecuzione. In allegato a questo documento sono forniti i file exp_10anni_pd.txt e exp_10anni_vr.txt.
<b>Testare il software col sistema ARPAV.</b>	Nessun errore rilevato.

Tabella 5.1. Pianificazione ed esito del test di collaudo.

## *Analisi dei requisiti*

**Nome del prodotto:**

Sistema di Esportazione Dati Alimenti per il Ministero della Salute  
(SEDAMS).

**Data:**

28/04/06

**Versione:**

1.0

**Stato del documento:**

Formale esterno

**Redazione:**

Piero Dalle Pezze

**Revisione:**

Piero Dalle Pezze    Padova, li 28/04/06    Firma .....

**Approvazione:**

Paolo Zambotto    Padova, li 28/04/06    Firma .....

**Lista di distribuzione:**

1. Paolo Zambotto
2. Piero Dalle Pezze

**Registro delle modifiche:**

v 1.0 28/04/06

Revisione formale del documento.

v 0.4 21/04/06

Integrazione paragrafo 3. Verifica del documento come descritto nel paragrafo 2.1 del Piano di qualifica.

v 0.3 20/04/06

Redazione dei paragrafi 3 e 4. Inserimento diagrammi Use Case. Aggiunta della figura 3.3.

v 0.2 19/04/06

Redazione dei paragrafi 1 e 2.

v 0.1 18/04/06

Creazione del documento.

## Sommario:

Questo documento presenta l'analisi del progetto *SEDAMS*, svolta dallo stagista Dalle Pezze Piero per l'azienda ARPAV

## Indice:

1. Introduzione	95
1.1. Scopo del documento	95
1.2. Scopo del prodotto	95
1.3. Glossario	95
1.4. Allegati	95
2. Descrizione generale	96
2.1. Contesto d'uso del prodotto	96
2.1.1. Processi produttivi e protocolli d'ambiente coinvolti	96
2.1.2. Piattaforma d'esecuzione ed interfacciamento con strumenti esterni	97
2.2. Funzioni del prodotto	97
2.3. Caratteristiche degli utenti	97
2.4. Vincoli generali	97
2.5. Assunzioni e dipendenze	97
3. Diagrammi Use Case	98
3.1. Diagramma Use Case generale	98
3.2. Diagramma Use Case di dettaglio della transazione	103
3.3. Diagramma Use Case delle statistiche	107
4. Lista dei requisiti	110
4.1. Requisiti funzionali	110
4.2. Requisiti di qualità	112
4.3. Requisiti di interfacciamento	112

# ***1. Introduzione***

## ***1.1. Scopo del documento***

Scopo di questo documento è di presentare l'analisi del sistema e del software SEDAMS illustrando, mediante tabelle e diagrammi UML, la cui documentazione è reperibile nel sito <http://www.omg.org>, i requisiti di sistema e software.

## ***1.2. Scopo del prodotto***

Lo scopo del prodotto è quello di esportare dati nella codifica richiesta dal Ministero della Salute specificata nel documento SA-99-D1006-015.

## ***1.3. Glossario***

Il glossario viene fornito come documento allegato ed è unico per tutta la documentazione.

## ***1.4. Allegati***

I seguenti documenti sono in allegato a questo:

- codice\_esportazione.pdf

Tale documento contiene la tabella di come deve essere codificato il file da esportare. Sarà citato nel requisito funzionale RFA11 descritto nel [paragrafo 4.1](#).

## ***2. Descrizione generale***

Il progetto *SEDAMS* consentirà all'azienda ARPAV di esportare i dati riguardanti le analisi effettuate su campioni di alimenti, presenti sia nel database regionale che nei database provinciali, nel formato specificato nel documento *SA-99-D1006-015* come specificato nel [paragrafo 1.2](#). Il progetto *SEDAMS* ha quindi l'obiettivo di automatizzare l'esportazione con gli immediati effetti di correttezza dell'invio di informazioni, di rapidità, di stabilità, oltre ovviamente al risparmio economico e lavorativo di personale incaricato altrimenti per questo compito.

### ***2.1. Contesto d'uso del prodotto***

Il flusso logico delle informazioni è il seguente:

1. Nel luogo di origine, le stazioni di monitoraggio, prelevano i dati.
2. I prelievi passano attraverso un livello di aggregazione ed integrazione provinciale.
3. I prelievi, quindi, arrivano a costituire la Banca Dati Ambientale, integrata di livello regionale.
4. Dalle banche dati regionali, il flusso informativo viene trasmesso agli organismi nazionali ed al pubblico.

Il progetto *SEDAMS*, si occuperà di prelevare le informazioni dai sette database provinciali dell'ARPAV a partire dalla postazione base regionale.

#### ***2.1.1. Processi produttivi e protocolli d'ambiente coinvolti***

Nessun processo produttivo o protocollo d'ambiente coinvolto.



### ***2.1.2. Piattaforma d'esecuzione ed interfacciamento con strumenti esterni***

Il software sarà eseguito dal personale incaricato per l'invio dei campioni all'organismo nazionale il quale quindi dovrà possedere i diritti di accesso ai database. Lo scopo del prodotto prevede quindi l'interfacciamento con il database Oracle principale dell'ARPAV e con i database Oracle provinciali della regione Veneto.

## ***2.2. Funzioni del prodotto***

Il prodotto deve consentire al responsabile per l'invio dei campioni e di analisi, di ottenere un file contenente l'insieme dei campioni e le analisi effettuate per ognuno di queste, codificato come previsto nel [paragrafo 1.2](#). I dati da esportare dovranno appartenere al modello 'ALIMENTI' ed il loro stato deve essere approvato. Il prodotto può essere reperibile in rete soltanto da personale autorizzato, ossia da coloro che possiedono l'accesso diretto ai database dell'azienda ARPAV

## ***2.3. Caratteristiche degli utenti***

L'utente che utilizzerà questo prodotto è l'incaricato per l'esportazione dei dati al Ministero della Salute o chi ne fa le veci.

## ***2.4. Vincoli generali***

Espressi nei requisiti di qualità

## ***2.5. Assunzioni e dipendenze***

Si suppone che l'applicazione venga installata nel server regionale dell'azienda ARPAV con sede a Padova. Nel server deve essere installato un web server e il database regionale dell'ARPAV

### 3. Diagrammi Use Case

Questo paragrafo presenta i diagrammi use case individuati per questo progetto e i rispettivi use case narrativi.

#### 3.1. Diagramma Use Case generale

La figura 3.1 illustra le funzioni base del software *SEDAMS*.

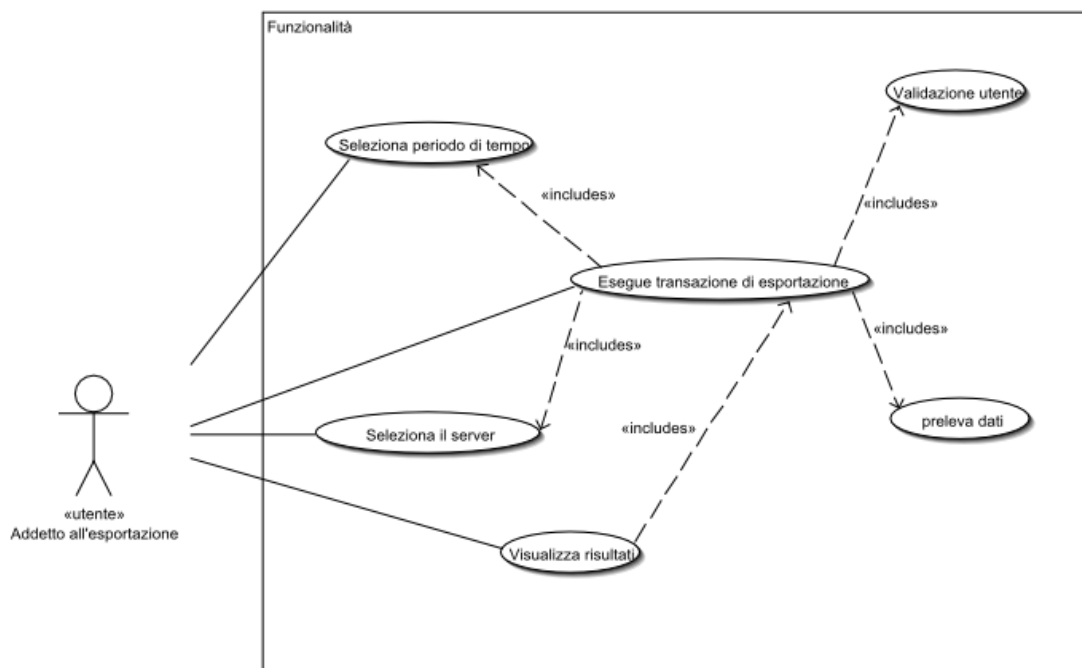


Figura 3.1. Diagramma Use Case del funzionamento di SEDAMS.

Di seguito sono presentati i rispettivi use case narrativi.

#### Use case: "Seleziona il server"

**Sigla:**

UC01.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

L'utente, deve decidere a quale server provinciale dell'ARPAV vuole connettersi. L'accesso al server consentirà all'utente di accedere al database ARPAV di quella provincia. Requisiti Funzionali: RFA3, RFA4.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. Di default, il server provinciale selezionato è DBPDLIMS, ossia il server di Padova dell'azienda ARPAV

**Flusso base di eventi:**

L'utente seleziona il server provinciale al quale desidera connettersi.

**Flussi alternativi:**

Nessuno.

**Post-condizioni:**

Il server provinciale è stato selezionato.

**Use case: "Seleziona il periodo di tempo"**

**Sigla:**

UC02.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

L'utente, deve decidere l'intervallo di tempo entro il quale vuole esportare i dati. Requisiti Funzionali: RFA1, RFA2.

**Pre-condizioni:**

L'utente deve essere all'interno del server regionale dell'azienda ARPAV

**Flusso base di eventi:**

L'utente seleziona in due appositi riquadri, la data di inizio e la data di fine.

**Flussi alternativi:**

Nessuno.

**Post-condizioni:**

Le date di inizio e di fine sono state selezionate.

**Use case: "Validazione utente"**

**Sigla:**

UC03.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

L'utente si autentica e si connette al server provinciale dal quale dovrà prelevare i dati. Requisiti Funzionali: RFA6, RFA13.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente ha attivato lo Use Case UC01.

**Flusso base di eventi:**

1. L'utente si è autenticato al server provinciale.
2. L'utente è connesso al server provinciale.

**Flussi alternativi:**

- Se il server selezionato in UC01 non esiste, l'utente è avvisato con un messaggio di errore.
- Se l'utente non dispone di un login di accesso al server, non vi accede e viene avvisato con un messaggio di errore.

**Post-condizioni:**

L'utente è autenticato e connesso al server provinciale.

### **Use case: "Esegue transazione di esportazione"**

**Sigla:**

UC04.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

L'utente avvia la transazione di esportazione dei dati dal database provinciale. Requisiti Funzionali: RFA5 – RFA12.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato gli Use Case UC01, UC02, UC03.

**Flusso base di eventi:**

1. L'utente avvia il processo transazionale di esportazione dei dati.
2. L'applicazione preleva i campioni e le rispettive analisi.
3. L'applicazione esegue l'esportazione generando il file di esportazione.
4. L'esportazione è conclusa.
5. L'utente è avvisato del termine dell'esportazione.
6. L'utente può selezionare il file creato ed aprirlo.

**Flussi alternativi:**

1. Se la procedura non può essere attivata, l'utente viene avvisato con un opportuno messaggio. Ciò avviene se le precondizioni non sono soddisfatte.

**Post-condizioni:**

Il file di esportazione è stato creato.

### **Use case: "Preleva dati"**

**Sigla:**

UC05.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

L'applicativo preleva i dati dal database provinciale.

Requisiti Funzionali: RFA12.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve essere stato autenticato e connesso al server provinciale.

**Flusso base di eventi:**

L'applicativo preleva i dati secondo il requisito funzionale RFA12.

**Flussi alternativi:**

Nessuno.

**Post-condizioni:**

I dati sono stati prelevati.

**Use case: "Visualizza risultati"****Sigla:**

UC06.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

L'utente, dopo aver esportato i dati nel formato richiesto decide di visualizzare il file generato. Requisiti Funzionali: RFA9, RFA10.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato lo Use Case UC04 correttamente.

**Flusso base di eventi:**

1. L'utente seleziona il file generato di esportazione.
2. L'utente apre il file.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessun file è generato.

**Post-condizioni:**

L'utente ha aperto il file.

### 3.2. Diagramma Use Case di dettaglio della transazione

La figura 3.2 illustra il meccanismo di esportazione transazionale del software SEDAMS con maggior dettaglio.

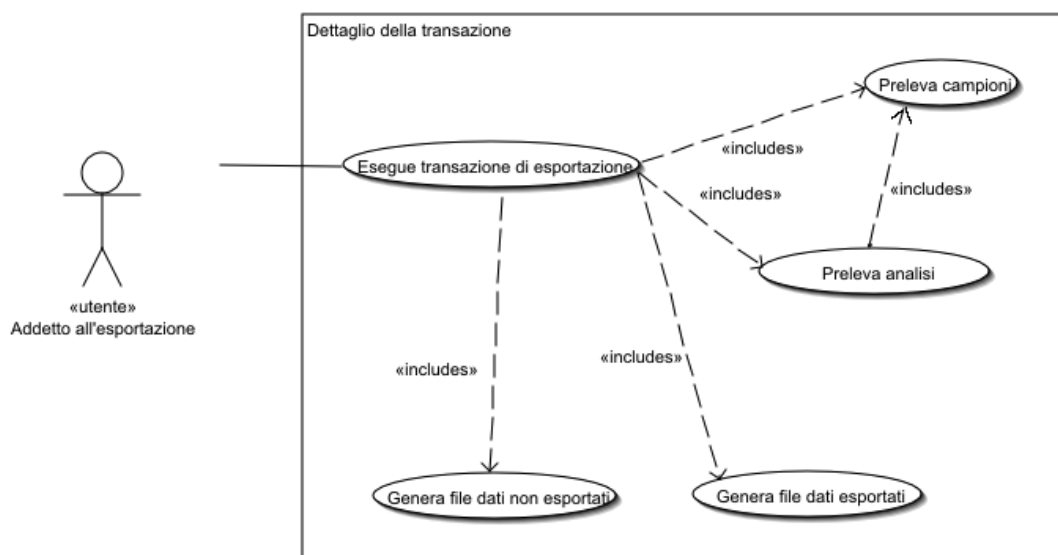


Figura 3.2. Diagramma Use Case della transazione.

Di seguito sono presentati i rispettivi use case narrativi.

### **Use case: "Preleva campioni"**

**Sigla:**

UC07.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

Il software inizia il prelevamento dei campioni degli alimenti autorizzati. Requisiti Funzionali: RFA12.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato lo Use Case UC03 correttamente e avviato lo Use Case UC04.

**Flusso base di eventi:**

1. Il software inizia a prelevare i campioni di tipo Alimenti, autorizzati.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessun prelevamento è effettuato.

**Post-condizioni:**

Il software ha prelevato i campioni.

### **Use case: "Preleva analisi"**

**Sigla:**

UC08.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

Il software inizia il prelevamento delle analisi autorizzate dei campioni degli alimenti autorizzati. Requisiti Funzionali: RFA12.

**Pre-condizioni:**



1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato lo Use Case UC04 correttamente.
3. Il software deve aver avviato lo Use Case UC07. Deve quindi essere prelevato già il campione. Qui si prelevano le analisi per il campione.

**Flusso base di eventi:**

1. Il software inizia a prelevare le analisi autorizzate per il campione autorizzato di tipo Alimenti.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessun prelevamento è effettuato.

Se la terza precondizione non è soddisfatta, nessun prelevamento di analisi è effettuato.

**Post-condizioni:**

Il software ha prelevato le analisi per quel campione.

**Use case: "Genera file dati esportati"**

**Sigla:**

UC09.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

Il software, dopo aver prelevato i campioni autorizzati di tipo Alimenti e le rispettive analisi autorizzate, esporta le informazioni nel formato richiesto. Il file di esportazione conterrà queste informazioni.

Requisiti Funzionali: RFA8, RFA11.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato lo Use Case UC04 correttamente.

3. Il software deve aver attivato lo Use Case UC07 e UC08.

**Flusso base di eventi:**

Il software genera il file di esportazione rispettando la codifica richiesta.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessun file è generato.

Se non ci sono dati da esportare, il software restituisce un file vuoto.

**Post-condizioni:**

Il software ha generato il file di esportazione.

**Use case: “Genera file dati non esportati”**

**Sigla:**

UC10.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

Il software, dopo aver prelevato i campioni autorizzati di tipo Alimenti e le rispettive analisi autorizzate, scrive in un file i campioni e le analisi che non sono state esportate per incompletezza o scorrettezza di valori. Requisiti Funzionali: RFB1, RFB2.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato lo Use Case UC04 correttamente.
3. Il software deve aver attivato lo Use Case UC07 e UC08.

**Flusso base di eventi:**

Il software genera il file di esportazione rispettando la codifica richiesta.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessun file è generato.

Se non ci sono anomalie, il software restituisce un file vuoto.

**Post-condizioni:**

Il software ha generato il file dei dati non esportati.

### 3.3. Diagramma Use Case delle statistiche

La figura 3.3 illustra la generazione e la visualizzazione delle statistiche nel software SEDAMS.

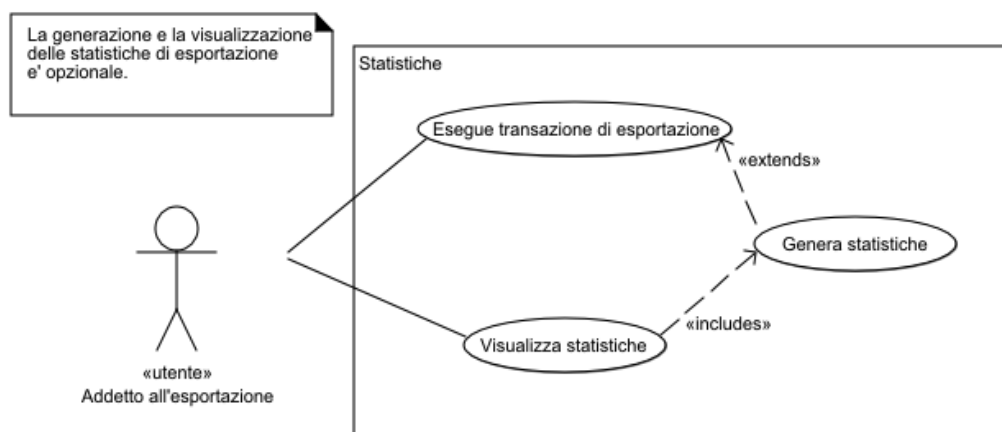


Figura 3.3. Diagramma Use Case della generazione e visualizzazione delle statistiche.

Di seguito sono presentati i rispettivi use case narrativi.

**Use case: “Genera statistiche”**

**Sigla:**

UC11.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

Il software, mentre genera il file di esportazione, genera anche delle statistiche sull'esportazione. Requisiti Funzionali: RFC1.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. L'utente deve aver attivato lo Use Case UC09.

**Flusso base di eventi:**

Il software genera le statistiche specificate in RFC1.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessuna statistica è generata.

**Post-condizioni:**

Il software, dopo aver completato l'esportazione, ha completato anche la generazione delle statistiche.

**Use case: "Visualizza statistiche"**

**Sigla:**

UC12.

**Attori coinvolti:**

Addetto all'esportazione.

**Scopo e Descrizione Sintetica:**

Il software, dopo aver generato le statistiche, le stampa a video.  
Requisiti Funzionali: RFC2.

**Pre-condizioni:**

1. L'utente deve essere all'interno del server regionale dell'azienda ARPAV
2. Lo Use Case UC11 deve essere stato attivato correttamente.

**Flusso base di eventi:**

Il software visualizza le statistiche generate a video.

**Flussi alternativi:**

Se la seconda pre-condizione non è soddisfatta, nessuna statistica è visualizzata.

**Post-condizioni:**

Il software ha visualizzato su video le statistiche generate dell'esportazione.

## 4. Lista dei requisiti

I paragrafi seguenti descrivono le liste dei requisiti funzionali, di qualità e di interfacciamento. I requisiti funzionali sono distinti con la sigla RF, quelli di qualità con la sigla RQ e quelli di interfacciamento con la sigla RI.

Ogni requisito inoltre appartiene ad una ed una sola classe di priorità tra le seguenti: obbligatorio, desiderabile e opzionale. Queste tre classi di requisiti saranno identificate rispettivamente con le sigle A (per i requisiti obbligatori), B (per i requisiti desiderabili) e C (per i requisiti opzionali).

La sintassi formale del nome di un requisito è la seguente:

tipo + classe di priorità + numero di requisito

Per esempio il requisito funzionale obbligatorio numero 5 è identificato con il nome univoco RFA5.

### 4.1. Requisiti funzionali

La tabella 4.1 elenca i requisiti funzionali obbligatori.

<b>Nome</b>	<b>Descrizione</b>
RFA1	Dalla GUI, l'utente deve poter impostare il periodo entro il quale vuole esportare i campioni alimentari autorizzati. In particolare deve poter selezionare una data di inizio ed una di fine. Queste date, quindi, rappresentano un filtro nella ricerca.
RFA2	La data di inizio deve essere precedente alla data di fine.
RFA3	Dalla GUI, l'utente deve poter impostare il nome del server provinciale dal quale intende prelevare i dati per ottenere il file di esportazione.
RFA4	Di default il server provinciale selezionato sarà quello di Verona (DBVRLIMS).
RFA5	L'utente avvia l'esportazione mediante un bottone di avvio,
RFA6	Se il requisito RFA2 non è soddisfatto, l'esportazione non sarà avviata e l'utente sarà avvisato con un messaggio di errore.
RFA7	L'esportazione deve avvenire in modo transazionale.

<b>Nome</b>	<b>Descrizione</b>
RFA8	L'esportazione deve produrre il file di esportazione richiesto.
RFA9	L'utente deve essere avvisato del termine dell'esportazione con un messaggio a video.
RFA10	Al termine dell'esportazione, deve essere disponibile nella pagina, il collegamento ipertestuale al file esportato.
RFA11	Il file esportato deve essere codificato secondo le specifiche espresse nel documento SA-99-D1006-015 che è stato formalizzato dallo stagista nel documento codifica_esportazione.pdf. In tale documento, nella colonna Note, sono presenti i vincoli che i campi devono rispettare.
RFA12	Il software dovrà prelevare ed esportare solo i campioni di tipo Alimenti e le analisi effettuate su questi campioni. Campioni e analisi devono essere stati autorizzati.
RFA13	Prima che la transazione di esportazione sia avviata, l'utente deve essersi autenticato al server provinciale.

Tabella 4.1. Requisiti funzionali obbligatori.

La tabella 4.2 elenca i requisiti funzionali desiderabili.

<b>Nome</b>	<b>Descrizione</b>
RFB1	L'esportazione produrrà anche il file contenente i campioni che non sono stati esportati perché non ammissibili.
RFB2	L'esportazione produrrà anche il file contenente le analisi che non sono state esportate perché non ammissibili.
RFB3	Al termine dell'esportazione deve essere disponibile il collegamento ipertestuale al file descritto nel requisito RFB1.
RFB4	Al termine dell'esportazione deve essere disponibile il collegamento ipertestuale al file descritto nel requisito RFB2.

Tabella 4.2. Requisiti funzionali desiderabili.

La tabella 4.3 elenca i requisiti funzionali opzionale.

<b>Nome</b>	<b>Descrizione</b>
RFC1	Mentre avanza la transazione, vengono generate anche le seguenti statistiche relative all'esportazione: <ol style="list-style-type: none"> <li>1. il numero dei campioni esportati;</li> <li>2. il numero dei campioni non esportati;</li> <li>3. il numero delle analisi esportate;</li> <li>4. la durata dell'esportazione;</li> <li>5. la dimensione del file.</li> </ol>

<b>Nome</b>	<b>Descrizione</b>
RFC2	Al termine dell'esportazione, l'utente potrà leggere le seguenti statistiche generate, specificate in RFC1.

Tabella 4.3. Requisiti funzionali opzionali.

## 4.2. *Requisiti di qualità*

La tabella 4.4 elenca i requisiti di qualità obbligatori.

<b>Nome</b>	<b>Descrizione</b>
RQA1	Il software <i>SEDAMS</i> deve essere realizzato utilizzando i linguaggi di programmazione PHP e PL/SQL di Oracle.
RQA2	Il nome del file contenente i dati da esportare deve essere quello specificato nel documento <i>SA-99-D1006-015</i> .

Tabella 4.4. Requisiti di qualità obbligatori.

La tabella 4.5 elenca i requisiti di qualità desiderabili.

<b>Nome</b>	<b>Descrizione</b>
-----	-----

Tabella 4.5. Requisiti di qualità desiderabili.

La tabella 4.6 elenca i requisiti di qualità opzionali.

<b>Nome</b>	<b>Descrizione</b>
-----	-----

Tabella 4.6. Requisiti di qualità opzionali.

## 4.3. *Requisiti di interfacciamento*

La tabella 4.7 elenca i requisiti di interfacciamento obbligatori.



<b>Nome</b>	<b>Descrizione</b>
RIA1	L'utente deve scegliere il server provinciale dal quale desidera effettuare l'esportazione, mediante una .
RIA2	Il software deve interfacciarsi con il database Oracle dell'azienda ARPAV.
RIA3	Chi accede al software, deve avere i permessi per accedere all'area riservata raggiungibile dal sito dell'azienda ARPAV.
RIA4	Il software sarà testato sulle macchine dell'azienda ARPAV.
RIA5	Il software dovrà essere intuitivo ed user-friendly.
RIA6	L'utente deve avere la possibilità di richiamare la funzionalità di aiuto. Tale pagina dovrà contenere tutte le informazioni utili e necessarie per comprendere il funzionamento del software <i>SEDAMS</i> .

Tabella 4.7. Requisiti di interfacciamento obbligatori.

La tabella 4.8 elenca i requisiti di interfacciamento desiderabili.

<b>Nome</b>	<b>Descrizione</b>
RIB1	L'utente seleziona la data mediante un "calendario grafico".
RIB2	Il software è realizzato utilizzando software Open Source.

Tabella 4.8. Requisiti di interfacciamento desiderabili.

La tabella 4.9 elenca i requisiti di interfacciamento opzionali.

<b>Nome</b>	<b>Descrizione</b>
-----	-----

Tabella 4.9. Requisiti di interfacciamento opzionali.

## *Specifica tecnica*

**Nome del prodotto:**

Sistema di Esportazione Dati Alimenti per il Ministero della Salute  
(SEDAMS).

**Data:**

25/05/06

**Versione:**

1.0

**Stato del documento:**

Formale esterno

**Redazione:**

Piero Dalle Pezze

**Revisione:**

Piero Dalle Pezze    Padova, li 25/05/06    Firma .....

**Approvazione:**

Paolo Zambotto    Padova, li 25/05/06    Firma .....

**Lista di distribuzione:**

1. Paolo Zambotto
2. Piero Dalle Pezze

**Registro delle modifiche:**

v 1.0 25/05/06

Revisione formale del documento.

v 0.5 04/05/06

Inserita tabella 4.3 nel capitolo 4.

v 0.4 03/05/06

Inserito diagrammi nel capitolo 2. Modificate tabelle nel capitolo 4.

v 0.3 02/05/06

Redatto capitolo 4 e 5.

v 0.2 01/05/06

Aggiunto paragrafi all'interno del capitolo 3.

v 0.1 29/05/06

Creazione del documento.

## Sommario:

Questo documento formalizza la progettazione architettuale del software *SEDAMS*.

## Indice:

1. Introduzione	117
1.1. Scopo del documento	117
1.2. Scopo del prodotto	117
1.3. Glossario	117
2. Definizione di prodotto	118
2.1. Metodo e formalismo di specifica	118
2.2. Primo livello di decomposizione architettuale	118
3. Descrizione dei singoli componenti	123
3.1. Interfaccia grafica per l'utente	123
3.2. Logica applicativa	124
3.2.1. Servizi del software SEDAMS	124
3.3. Memorizzazione dei dati	125
4. Stime di fattibilità e di bisogno di risorse	126
5. Tracciamento della relazione componenti – requisiti	128

# ***1. Introduzione***

## ***1.1. Scopo del documento***

Presentare l'architettura astratta del software *SEDAMS*, utilizzando rigore e diagrammi formali UML.

## ***1.2. Scopo del prodotto***

Lo scopo del prodotto è quello di esportare dati nella codifica richiesta dal Ministero della Salute specificata nel documento *SA-99-D1006-015*.

## ***1.3. Glossario***

Il glossario viene fornito come documento allegato ed è unico per tutta la documentazione.

## ***2. Definizione del prodotto***

### ***2.1. Metodo e formalismo di specifica***

Questo documento, intende utilizzare il linguaggio di progettazione UML per identificare l'architettura astratta del software *SEDAMS*. Si utilizzeranno, quindi, i diagrammi use case, descritti nel documento Analisi dei requisiti, per formalizzare una visione compatta e globale del sistema. Per fare ciò, si adotteranno i diagrammi di deployment, delle classi di sequenza e delle collaborazioni.

### ***2.2. Primo livello di decomposizione architetturale***

I server dell'azienda ARPAV sono strutturati come descritto in figura 2.1, che presenta il diagramma di deployment dell'architettura globale. Nel diagramma si può osservare che l'azienda ARPAV è partizionata in sette server regionali caratterizzati dal nome così formalizzato:

database + provincia + lims

Per esempio, il database della provincia di Padova è identificato con il nome DBPDLIMS. I server provinciali sono monitorati e controllati dal server regionale (con sede a Padova), il quale, quindi rappresenta il Veneto, nell'azienda complessiva dell'ARPA.

Il software *SEDAMS* sarà installato all'interno del server regionale dell'ARPAV e si conatterà al server provinciale specificato. La motivazione per la quale il software non può interfacciarsi direttamente con il server regionale per esportare tutti i dati delle sette provincie del Veneto, è dovuta alla mancanza della reperibilità di un consistente numero di dati in tale server.

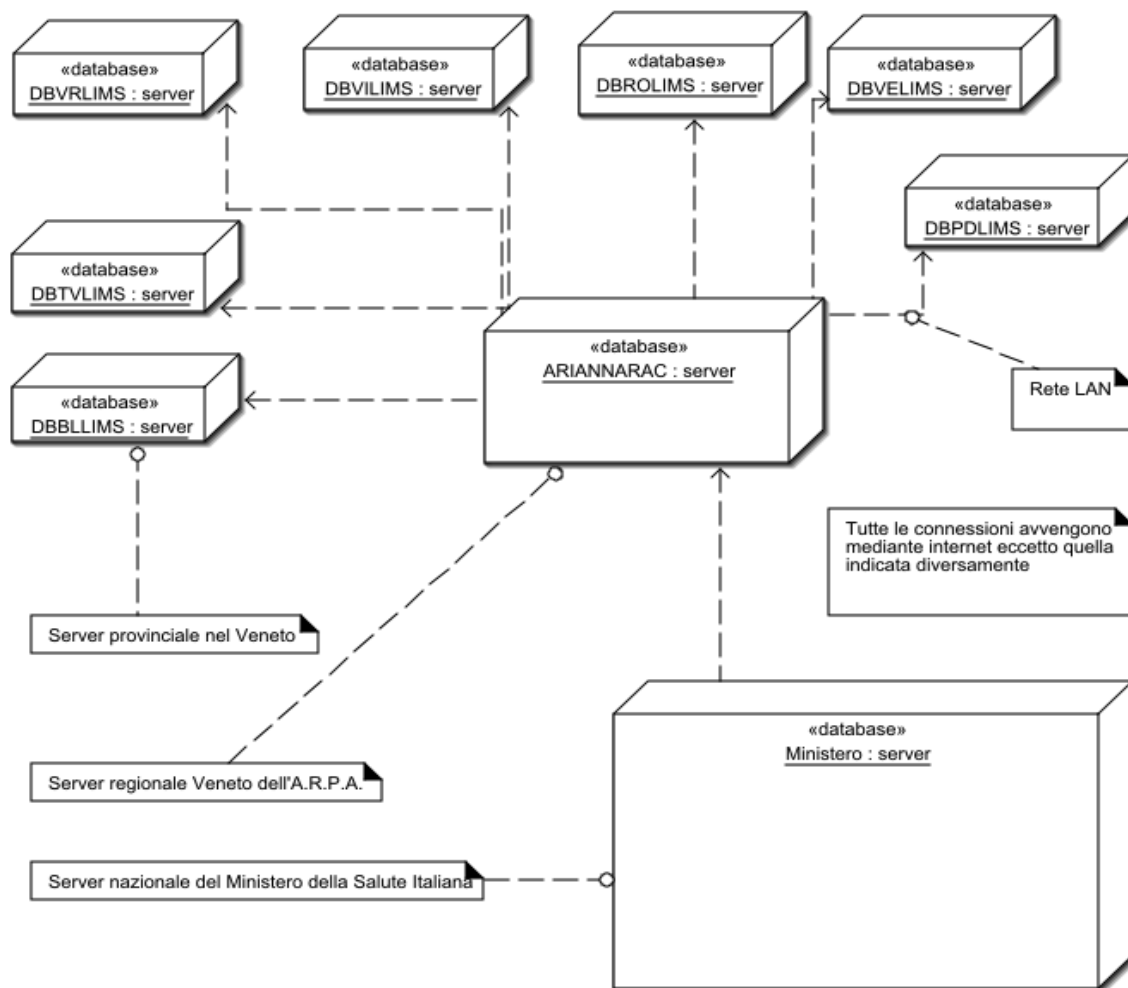


Figura 2.1. Diagramma di deployment dei server ARPAV.

Nella figura 2.2, si presenta il diagramma delle classi dell'architettura astratta del software SEDAMS. Si può osservare immediatamente che si è preso in forte considerazione il design pattern denominato *three-tier*. In questo si può osservare come l'utente si interfaccia alla GUI, la quale richiamerà il servizio che il software garantisce.

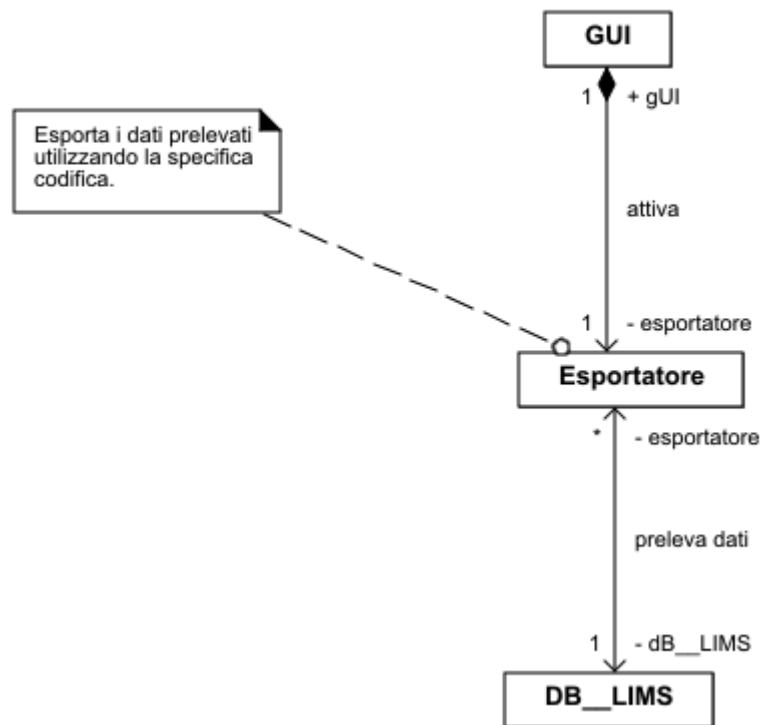


Figura 2.2. Diagramma delle classi del software SEDAMS.

Nella figura 2.3, si presenta il diagramma di sequenza del software *SEDAMS*. Fondamentalmente, sono state considerate due possibilità progettuali:

1. effettuare il prelievo e trasporto di tutti i dati necessari, dal server provinciale al server centrale, utilizzando delle tabelle provvisorie. Poi, esportare i dati nella nuova codifica;
2. prelevare tutti i campioni immediatamente e poi, per ciascuno di questi, prelevare le rispettive analisi. Una volta ottenute le analisi, si procederà alla scrittura del campione e delle sue analisi nel file di esportazione senza l'utilizzo di tabelle aggiuntive.



Dopo una discussione, con il tutor aziendale, si è stabilito di scegliere la seconda progettazione per i seguenti due motivi:

1. L'applicazione sarà impiegata periodicamente e non frequentemente.
2. L'azienda ARPAV intende nell'imminente futuro, connettere il software direttamente nel server regionale e da questo, prelevare tutti i dati dalle sette province del Veneto in una sola volta. Con la seconda progettazione e con queste intenzioni, si risparmieranno sicuramente tempo e spazio, in quanto non sarà necessario l'utilizzo di tabelle intermedie e la riscrittura dei dati.

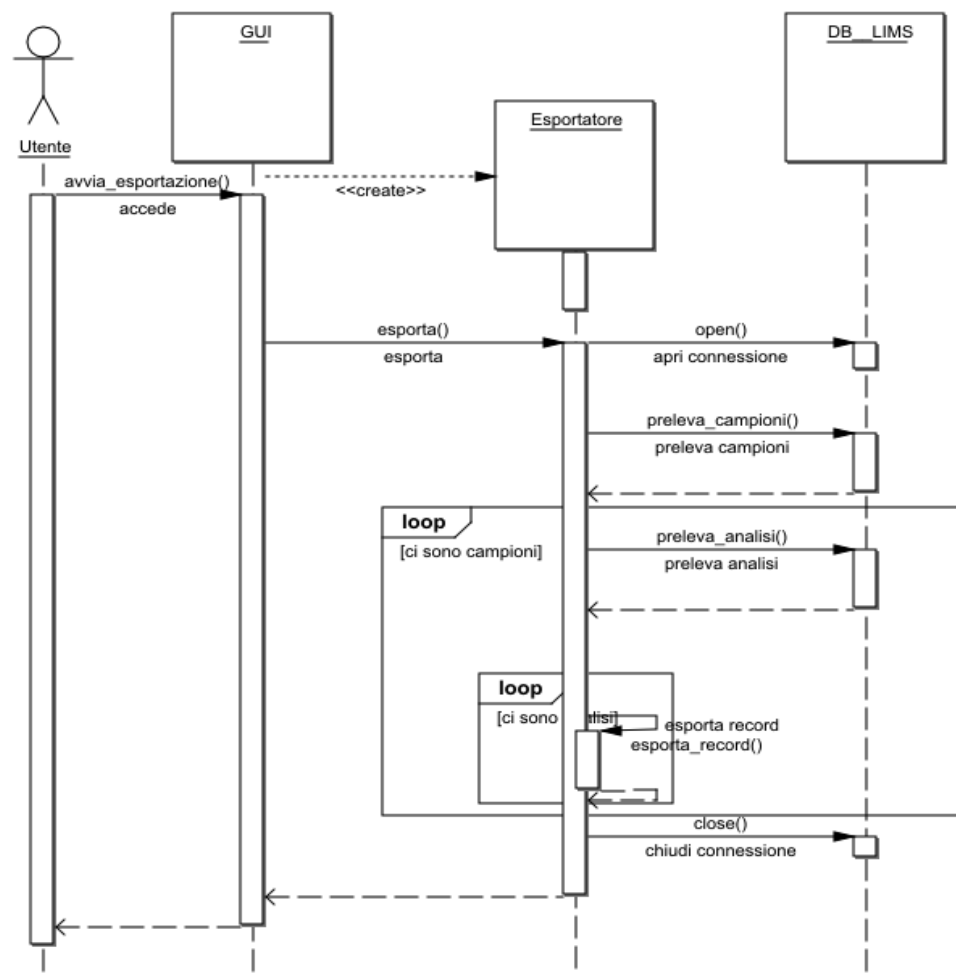


Figura 2.3. Diagramma di sequenza del software SEDAMS.

Nella figura 2.4, è presentata la vista rappresentante la collaborazione tra gli oggetti. Tale diagramma è stato costruito a partire dal precedente diagramma UML di sequenza.

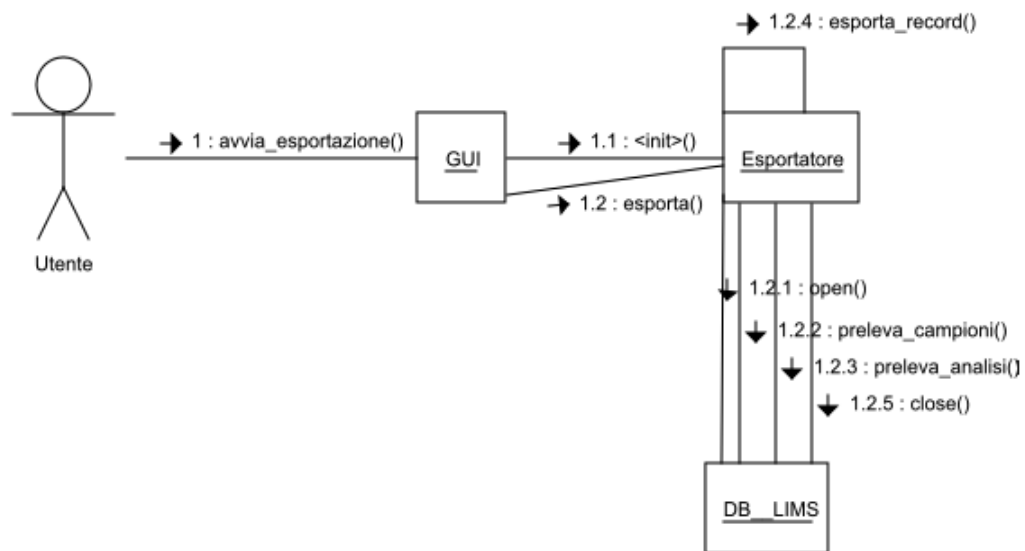


Figura 2.4. Diagramma di collaborazione del software *SEDAMS*.

### ***3. Descrizione dei singoli componenti***

Le quattro classi rappresentano di fatto le componenti del design pattern architetturale *three-tier*. Di seguito si fornisce una descrizione.

#### ***3.1. Interfaccia grafica per l'utente***

##### **Tipo, obiettivo e funzione della classe/interfaccia**

Fornisce un'interfaccia grafica mediante la quale l'utente può utilizzare il servizio del software *SEDAMS*. Sull'interfaccia grafica l'utente specificherà i parametri descritti nel documento [Analisi dei requisiti](#) e potrà visualizzare l'evoluzione della transazione, l'esito e il prodotto.

##### **Relazioni d'uso di altre classi/interfacce**

Si interfaccia con il livello della logica applicativa, ossia con l'Esportatore.

##### **Interfacce e relazioni di uso da altre classi/interfacce**

E' aggiornata direttamente dal livello della logica applicativa, ossia dall'Esportatore, nel momento in cui l'evoluzione della transazione si modifica.

##### **Attività svolte e dati trattati**

Le attività svolte da questa componente sono:

- Inviare alla logica applicativa i parametri inseriti dall'utente per esportare i dati.
- Visualizzare lo stato attuale dell'esportazione.

I dati trattati sono:

- I parametri inseriti dall'utente.
- Il link al file di esportazione prodotto.

## **3.2. Logica applicativa**

### **3.2.1. Servizi del software SEDAMS**

#### **Tipo, obiettivo e funzione della classe/interfaccia**

Generare il file di esportazione dati richiesto, nella codifica descritta nel documento Analisi dei requisiti. Definisce la transazione nel suo complesso. Fornisce quindi anche il servizio di accesso al database. In particolare, tutti i dati richiesti, devono essere prelevati, come specificato nel documento Analisi dei requisiti.

#### **Relazioni d'uso di altre classi/interfacce**

Si interfaccia con la GUI del software *SEDAMS* aggiornando la finestra dello stato della transazione e dell'esito finale. Si interfaccia con il database dal quale dovrà prelevare i dati richiesti.

#### **Interfacce e relazioni di uso da altre classi/interfacce**

Riceve i parametri dalla GUI, la quale lo attiva.

#### **Attività svolte e dati trattati**

Le attività svolte da questa componente sono:

- Prelevare i dati richiesti dal database. In particolare, provvede a prelevare in una sola occasione tutti i campioni di tipo 'ALIMENTI' che sono stati autorizzati. In seguito, preleva per ciascuno di questi, le analisi rispettive.
- Tradurre nella codifica specificata nel documento citato sopra, i dati precedentemente prelevati.
- Aggiornare la GUI sullo stato della transazione.
- Specificare il file di esportazione alla GUI.

I dati trattati sono:

- I parametri inseriti dall'utente.
- I dati da prelevare dal database specificati nel documento Analisi dei

requisiti. (Campioni ed Analisi di questi).

- L'oggetto GUI.

### **3.3. Memorizzazione dei dati**

#### **Tipo, obiettivo e funzione della classe/interfaccia**

Rappresenta il database DB\_\_LIMS, ossia un database di una generica provincia dell'azienda ARPAV. Contiene i dati richiesti dal software *SEDAMS*. Questa componente è già realizzata.

#### **Relazioni d'uso di altre classi/interfacce**

Non utilizza il software *SEDAMS*.

#### **Interfacce e relazioni di uso da altre classi/interfacce**

E' utilizzato dall'Esportatore della logica applicativa.

#### **Attività svolte e dati trattati**

Le attività svolte da questa componente sono:

- Ritornare le informazioni richieste alla logica applicativa.

I dati trattati sono:

- I dati specificati nel documento Analisi dei requisiti.

## 4. Stime di fattibilità e di bisogno di risorse

Nel documento Codifica di esportazione, allegato all'Analisi dei requisiti, sono analizzati tutti i campi che dovranno essere ricercati dal software *SEDAMS*. Esistono, tuttavia, dei campi dati richiesti per l'esportazione, che non sono ancora gestiti all'interno del database.

E' stato concordato in una riunione dell'azienda ARPAV che alcuni campi dovranno essere gestiti appropriatamente dai laboratori LIMS stessi, ossia, ogni provincia del Veneto si è assunta la responsabilità di gestire tali campi all'interno del proprio database. Esistono invece, altri campi, che vanno gestiti dallo stagista Dalle Pezze Piero.

Nella tabella 4.1, sono riportati i campi che ogni laboratorio LIMS dovrà gestire.

N°	Nome del campo	Gestione
27	reg_naz_prod	Deve essere riempita la colonna AV_NAZIONALITA della tabella SAMPLE con i codici delle regioni italiane o con la sigla della nazione.
40	cod_det	Deve essere riempito il campo AV_CAMPO1 della tabella COMPONENT. In futuro occorrerebbe spostare il contenuto della colonna AV_CAMPO1 nella colonna AV_MINISTERO per una maggiore coerenza di significato.

Tabella 4.1. Campi che i laboratori LIMS dovranno gestire.

Nella tabella 4.2, sono riportati i campi che lo stagista dovrà gestire.

N°	Nome del campo	Gestione
1, 2, 3, 4	lab_tipo, cod_reg, cod_usl, prog_lab	Tale tabella (Descrizione PMP) è già disponibile per il programma Access di Microsoft e contiene i codici PMP (concatenazione dei 4 campi) per le sette province della regione Veneto. Si può gestire facilmente con un array che contiene tali codici e selezionare il

<b>N°</b>	<b>Nome del campo</b>	<b>Gestione</b>
		codice in base al server provinciale cui ci si connette.
41	tecn_anal	Occorre importare la tabella TECNICHE_ANALITICHE data dal Ministero della Salute nei database LIMS. In seguito occorre prelevare il campo TAN_COD della tabella TECNICHE_ANALITICHE eseguendo un'operazione di join con il campo TECNICA_ANALITICA della tabella AV_DET_ALIMENTI. La tabella AV_DET_ALIMENTI deve essere filtrata in modo tale che il campo CODICE sia uguale al codice di determinazione alimento caratteristico dell'analisi che si sta prelevando.
42	unit_mis	Deve essere riempita la colonna AV_COD_MINISTERO della tabella UNITS con i codici delle unità del Ministero.

Tabella 4.2. Campi che lo stagista Dalle Pezze Piero dovrà gestire.

Il software è concepito per esportare i dati che sono stati autorizzati nel periodo di 1 anno. Nella tabella 4.3, si forniscono delle stime di misurazione:

<b>Oggetto</b>	<b>Misura stimata</b>
Numero medio di campioni / anno	1300
Numero medio di analisi per campione / anno	40
Campi da prelevare per 1 campione	26
Campi da prelevare per 1 analisi di 1 campione	7
Numero medio di informazioni / anno	52000 (= 1300 campioni * 40 analisi)
Grandezza file di esportazione	104 MB (codifica ASCII) (= 52000 * 1 B * (2000 celle per informazione))
Durata della transazione prevista	45 minuti

Tabella 4.3. Stima delle misurazioni per il software SEDAMS.

## 5. Tracciamento della relazione componenti – requisiti

Nella tabella 5.1 è presente la matrice di tracciamento dei componenti con i requisiti specificati nel documento Analisi dei requisiti. Si può osservare che tutti i requisiti vengono tracciati nelle componenti definite in questo documento.

<b>MATRICE DI TRACCIAMENTO</b>	<b>GUI</b>	<b>Esportatore</b>	<b>DB_LIMS</b>
<b>RFA1</b>	x		
<b>RFA2</b>	x		
<b>RFA3</b>	x		
<b>RFA4</b>	x		
<b>RFA5</b>	x		
<b>RFA6</b>	x		
<b>RFA7</b>	x	x	
<b>RFA8</b>	x	x	
<b>RFA9</b>	x	x	
<b>RFA10</b>	x	x	
<b>RFA11</b>		x	x
<b>RFA12</b>		x	x
<b>RFA13</b>	x		
<b>RFB1</b>	x	x	
<b>RFB2</b>	x	x	
<b>RFB3</b>	x	x	
<b>RFB4</b>	x	x	
<b>RFC1</b>	x	x	
<b>RFC2</b>	x		

Tabella 5.1. Matrice di tracciamento componenti – requisiti.



## **Definizione di prodotto**

**Nome del prodotto:**

Sistema di Esportazione Dati Alimenti per il Ministero della Salute  
(SEDAMS).

**Data:**

25/05/06

**Versione:**

1.0

**Stato del documento:**

Formale esterno

**Redazione:**

Piero Dalle Pezze

**Revisione:**

Piero Dalle Pezze    Padova, lì 25/05/06    Firma .....

**Approvazione:**

Paolo Zambotto    Padova, lì 25/05/06    Firma .....

**Lista di distribuzione:**

1. Paolo Zambotto
2. Piero Dalle Pezze

**Registro delle modifiche:**

v 1.0 25/05/06

Revisione formale del documento.

v 0.3 23/05/06

Completamento capitolo 3. Stesura capitolo 4 e 5. Realizzato diagramma delle componenti.

v 0.2 15/05/06

Scrittura capitolo 2 e 3. Realizzato diagramma delle classi.

v 0.1 29/04/06

Creazione del documento.

## Sommario:

Fornire la specifica generale e di dettaglio del software *SEDAMS*.

## Indice:

1. Introduzione	132
1.1. Scopo del documento	132
1.2. Scopo del prodotto	132
1.3. Glossario	132
2. Standard di progetto	133
2.1. Standard di progettazione architettuale	133
2.2. Standard di documentazione del codice	133
2.3. Standard di denominazione di entità relazioni	134
2.4. Standard di programmazione	134
2.5. Standard di lavoro	135
3. Specifica delle componenti	136
4. Codice sorgente	146
5. Tracciamento della relazione componenti requisiti	148

# ***1. Introduzione***

## ***1.1. Scopo del documento***

Presentare formalmente l'architettura finale astratta e di dettaglio del software *SEDAMS* e fornire il suo codice sorgente.

## ***1.2. Scopo del prodotto***

Lo scopo del prodotto è quello di esportare dati nella codifica richiesta dal Ministero della Salute specificata nel documento *SA-99-D1006-015*.

## ***1.3. Glossario***

Il glossario viene fornito come documento allegato ed è unico per tutta la documentazione.

## ***2. Standard di progetto***

Nei paragrafi seguenti, sono presentati i principali standard adottati nel progetto *SEDAMS*.

### ***2.1. Standard di progettazione architetturale***

Nella definizione dell'architettura, si è utilizzato lo standard architetturale denominato *three-tier*. In particolare, sono presenti tre componenti: l'interfaccia grafica con la quale l'utente è in grado di interfacciarsi con il sistema, la componente di accesso ai e di trattamento dei dati ed infine, il database dal quale prelevare i dati. In questo contesto, lo sviluppo del software *SEDAMS*, costituisce nella realizzazione delle prime due componenti e dell'importazione di alcune tabelle per particolari codifiche ministeriali nel database centrale dell'azienda ARPAV.

### ***2.2. Standard di documentazione del codice***

Si è provveduto a documentare appropriatamente ogni intestazione di classe, metodo, funzione, campo dato, variabile globale utilizzata ed il codice interno di funzioni complesse. Si è stabilito di documentare ogni intestazione utilizzando lo stile:

```
/** commento */
```

in modo tale che, con l'ausilio di appositi strumenti di generazione automatica di documentazione per il linguaggio di programmazione PHP, non servissero ulteriori modifiche.

I campi dati della classe *EsportatoreAlimenti* che costituiscono i campi da esportare sono documentati con la seguente notazione:

CommentoCampoDaEsportare ::= "/"\*\* Campo " + numero + " : " + tipo + "." +  
Descrizione + "(" + byte + "B)." + ["(Valore di  
default)."] + " \*/"

Per esempio il campo cod\_strut sarà documentato come segue:

/\*\* Campo 5 : string. Il codice della struttura ARPA (1B). (Valore di default). \*/

### ***2.3. Standard di denominazione di entità e relazioni***

Il nome utilizzato per i campi dati ed i metodi è esattamente quello presentato nel documento Codifica esportazione, allegato al documento Analisi dei requisiti. I nomi riportati dovranno essere una riduzione del nome originale, ma di facile interpretazione, in modo tale che anche il personale dell'azienda ARPAV possa facilmente coglierne il significato.

### ***2.4. Standard di programmazione***

Ogni file deve contenere una singola classe, dalla quale prende il nome, questo ai fini di una buona suddivisione logica, e per evitare file lunghi e di difficile comprensione.

Per garantire una buona leggibilità, il codice deve essere ben formattato ed indentato. Questo compito, a carico dello stagista Dalle Pezze Piero. Si utilizzerà per tale scopo lo strumento di indentazione automatica fornito dall'editor stesso.

Non è imposta una lunghezza massima di riga di codice, dato che la formattazione automatica fornita dall'ambiente di sviluppo, provvede a dividere le righe troppo lunghe.

Eccetto il nome delle classi, per i quali si adotterà lo standard della prima lettera maiuscola, ogni altro nome, sia esso di metodo che di campo dato, sarà in minuscolo. Eventuali nomi che sono caratterizzati da più parole, saranno suddivisi da un carattere di underscore ("\_").

Le variabili che saranno inizializzate con i dati prelevati dal database, devono

avere il nome della colonna che rappresentano nel database. Ciò è indispensabile per una rapida consultazione del codice anche da parte di coloro che non hanno sviluppato il codice.

Tutte le modifiche apportate al codice, devono essere annotate nel registro delle modifiche posto nell'intestazione del file.

In fase di compilazione non devono essere rilevati errori fatali, o potenziali (warning).

## ***2.5. Standard di lavoro***

Non sono stati imposti particolari strumenti da parte dell'azienda ARPAV allo stagista Dalle Pezze Piero. Di conseguenza, lo stagista ha utilizzato preferibilmente strumenti Open Source quali l'editor di testo [XEmacs](#), il browser [Firefox Mozilla](#) e il software [TOAD](#) dell'azienda ARPAV Per lo sviluppo dei diagrammi UML, si è utilizzato il tool [EclipseUML](#).

### 3. Specifica delle componenti

Questo capitolo costituisce un affinamento del capitolo 3 del documento Specifica tecnica. In questi termini, quest'ultimo documento è completa il presente delineando l'architettura di alto livello del software *SEDAMS*.

La struttura dei file è presentata mediante il diagramma delle componenti nella figura 3.1. Eccetto i moduli `Esportatore` e `EsportatoreAlimenti`, per i quali si è adottata una progettazione ad oggetti, tutti gli altri sono di fatto progettati e realizzati utilizzando procedure, dal momento che rappresentano pagine web con al più un breve algoritmo di stampa.

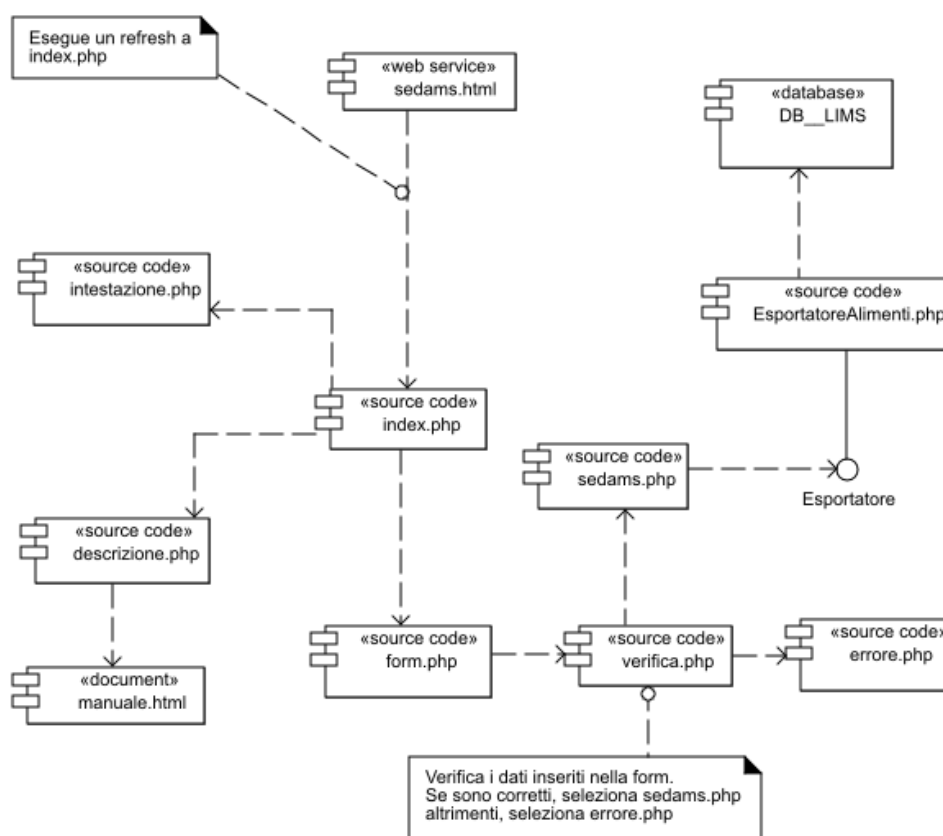


Figura 3.1. Diagramma delle componenti che illustra la struttura dei files.



### ***3.1. Interfaccia grafica per l'utente***

#### **Tipo, obiettivo e funzione della classe/interfaccia**

Fornisce un'interfaccia grafica mediante la quale l'utente può utilizzare il servizio del software *SEDAMS*. Sull'interfaccia grafica l'utente specificherà i parametri descritti nel documento [Analisi dei requisiti](#) e potrà visualizzare l'evoluzione della transazione, l'esito e il prodotto. L'interfaccia grafica è realizzata con il linguaggio HTML e suddivisa in diversi frame che costituiranno l'area di inserimento dati, l'area di visualizzazione dello stato attuale della transazione e l'area di visualizzazione dei collegamenti ipertestuali ai file di esportazione prodotti. I dati saranno inseriti mediante l'utilizzo di una form la quale invierà i dati alla classe *EsportatoreAlimenti*.

#### **Relazioni d'uso di altre classi/interfacce**

Si interfaccia con l'oggetto *EsportatoreAlimenti* al quale invia i dati inseriti nella form e dal quale ottiene l'output dello stato della transazione e del file di esportazione prodotto.

#### **Interfacce e relazioni di uso da altre classi/interfacce**

E' aggiornata direttamente dalla logica applicativa, ossia dalla classe *EsportatoreAlimenti*, nel momento in cui l'evoluzione della transazione si modifica.

#### **Attività svolte e dati trattati**

Le attività svolte da questa componente sono:

- Inviare alla classe *EsportatoreAlimenti* i parametri inseriti dall'utente per esportare i dati.
- Visualizzare lo stato attuale dell'esportazione.

I dati trattati sono di due tipi:

1. Parametri inseriti dall'utente:
  - 1.1. Login dell'utente.
  - 1.2. Password dell'utente.
  - 1.3. Server provinciale al quale ci si intende connettere. Deve essere

uno dei server provinciali presentati in figura 2.1 del documento Specifica tecnica.

- 1.4. La data a partire dalla quale si intende effettuare l'esportazione.
- 1.5. La data di fine esportazione.
- 1.6. Un bottone di avviamento dell'esportazione.
2. Il link al file di esportazione prodotto. Il nome del file da esportare al Ministero della Salute, secondo il documento *SA-99-D1006-015* è `eplafr01.dax`. I file provinciali prodotti avranno per cui il nome `eplafr01_SiglaProvinciaVeneta.dax`.

### **3.2. Logica applicativa**

Nella figura 3.2, è presentato il diagramma delle classi della logica applicativa. Di seguito, vengono analizzati i singoli moduli, rispettivamente `Esportatore` e `EsportatoreAlimenti`. Al loro interno è inoltre presentata la specifica di ogni campo dato e metodo impiegato. Per motivi di spazio, nel diagramma seguente, si è preferito tralasciare i parametri dei metodi. Nelle tabelle 3.1 e 3.2, questi vengono presentato e descritti nel dettaglio.



Figura 3.2. Diagramma delle classi che illustra l'architettura di dettaglio.

### **3.2.1. Esportatore**

#### **Tipo, obiettivo e funzione della classe/interfaccia**

Interfaccia di un generico esportatore dati.

#### **Relazioni d'uso di altre classi/interfacce**

Non utilizza direttamente classi e interfacce. Deve essere istanziato.

#### **Interfacce e relazioni di uso da altre classi/interfacce**

Si interfaccia con la GUI del software SEDAMS aggiornando la finestra dello stato della transazione e dell'esito finale. Si interfaccia con il database dal quale dovrà prelevare i dati richiesti.

#### **Attività svolte e dati trattati**

Definisce le caratteristiche di un generico esportatore dati.

I dati trattati sono:

- `esporta($filename, $filename_err)`. Funzione che si occupa del prelevamento e dell'esportazione dei dati nel file `$filename`. Nel file `$filename_err` sono inseriti i campioni che non sono stati esportati.

### **3.2.2. EsportatoreAlimenti**

#### **Tipo, obiettivo e funzione della classe/interfaccia**

Classe che implementa l'interfaccia `Esportatore`. Lo scopo è quello di generare il file di esportazione dati richiesto, nella codifica descritta nel documento [Analisi dei requisiti](#). Definisce la transazione nel suo complesso. Fornisce quindi anche il servizio di accesso al database. In particolare, tutti i dati richiesti, devono essere prelevati, come specificato nel documento [Analisi dei requisiti](#).

#### **Relazioni d'uso di altre classi/interfacce**

Si interfaccia con la GUI del software SEDAMS aggiornando la finestra dello stato della transazione e dell'esito finale. Si interfaccia con il database dal quale dovrà prelevare i dati richiesti.

## Interfacce e relazioni di uso da altre classi/interfacce

Riceve i parametri dalla GUI, la quale lo attiva.

## Attività svolte e dati trattati

Nella tabella 3.1 sono riportati i campi dati della classe:

Nome	Descrizione
<b>Comandi SQL</b>	
sql_preleva_campioni	Stringa di SELECT per il prelevamento dei campioni di tipo ALIMENTI autorizzati.
sql_preleva_analisi	Stringa di SELECT per il prelevamento delle analisi di un campione di tipo ALIMENTI autorizzato.
<b>Periodo di esportazione</b>	
data_ini	Definisce la data iniziale a partire dalla quale si intende esportare.
data_fin	Definisce la data finale entro la quale si intende esportare.
<b>Parametri di connessione</b>	
server	Il nome del server a cui ci si intende connettere.
login	La login dell'utente.
password	La password dell'utente.
<b>Campi dei campioni</b>	
cod_tipi_lab	Un array contenente i sette codici pmp delle provincie della regione Veneto.
cod_pmp	Il codice della provincia della regione veneto alla quale ci si connette. (8B)
cod_strut	Il Codice della struttura ARPA (1B). (Valore di default).
anno_camp	L'anno di revisione del campione (4B).
prog_camp	Il progressivo del campione (6B).
grado_istanza	Grado di istanza (1B). (Valore di default).
tipo_record	Tipo di <i>record</i> (2B)(Se < 9 aggiungere "0" davanti).(Campo del Ministero della Salute).
tipo_esame	Il tipo dell'esame che è stato fatto sul campione. L'esame può essere di tipo chimico (C), biologico (B) o fisico (F).
cod_alim_appart	Il codice dell'alimento (9B).
den_alim	La denominazione dell'alimento (60B).
data_prel	Una stringa contenente l'anno, il mese e il giorno del prelievo del campione nel formato YYYYMMDD(8B).
motivo_prel	Il motivo del prelievo (2B). (Valore di default).
cod_filiera_alim	Il codice della filiera alimentare (1B). (Valore di default).
cod_reg_prel	Il codice della regione nella quale è stato effettuato il prelievo (3B).

<b>Nome</b>	<b>Descrizione</b>
cod_prov_prel	Il codice della provincia nella quale è stato effettuato il prelievo (3B).
cod_com_prel	Il codice del comune nella quale è stato effettuato il prelievo (3B).
org_prelevatore	Il codice dell'organismo che ha effettuato il prelievo del campione (3B).
org_prelevatore_utente	Organismo prelevatore utente (3B). (Valore di default).
cod_fisc_prod	Il codice fiscale del produttore (16B). (Valore di default).
den_prod	La denominazione del produttore (40B).
reg_naz_prod	Il codice della regione del produttore se questo è in Italia, altrimenti il codice della nazione
cod_fisc_dist	Il codice fiscale del distributore (16B). (Valore di default).
den_dist	La denominazione del distributore (40B).
mod_conser	Il codice che rappresenta la modalità di conservazione del campione (2B).
tipo_conten	Il codice che rappresenta il tipo di contenitore utilizzato (2B).
lab_sec_istanza	La seconda istanza di laboratorio (3B). (Valore di default).
esito_globale	L'esito globale del campione (1B). (Valore di default).
note	Note di considerazioni (210B). (Attualmente valore di default).
data_fine_analisi	Stringa contenente l'anno, il mese ed il giorno in cui l'analisi è stata completata nel formato YYYYMMDD (8B).
controllo_ufficiale	Afferma se sull'analisi è stato eseguito un controllo ufficiale (2B). (Valore di default).
<b>Campi delle analisi</b>	
cod_det	Il codice di determinazione (4B).
prog_det	Il progressivo di determinazione (1B).
tecn_anal	Il codice della tecnica analitica impiegata (2B).
unit_mis	Il codice dell'unità di misura impiegata (3B).
flag_det	Il flag di determinazione (1B). (Valore di default).
flag_accr	Il flag di accreditamento (1B). (Valore di default).
unit_camp	L'unità campionaria (2B). (Valore di default).
segno	Il segno (1B).
ris_sing_det	Il risultato della singola determinazione (15B).
esito_analisi	L'esito dell'analisi (1B).
<b>Campi di chiusura</b>	
program_minister	Afferma se questo è un programma ministeriale (1B). (Valore di default).
data_inserimento	La data attuale (8B) (DDMMYYYY).
filler_rec_01	Stringa di blank di chiusura per i <i>record</i> di tipo 01 (32B).

<b>Nome</b>	<b>Descrizione</b>
	(Valore di default).
filler_rec_cont	Stringa di blank di chiusura per i <i>record</i> di continuazione (8B). (Valore di default).

Tabella 3.1. Campi dati della classe EsportatoreAlimenti.

Nella tabella 3.2 sono contenuti i metodi della classe (la modalità del settaggio è definita nel documento *SA-99-D1006-015*) :

<b>Nome</b>	<b>Descrizione</b>
EsportatoreAlimenti(\$data_inizio,\$data_fine, \$server, \$login, \$password)	Inizializza i campi dati ricevuti e quelli di default secondo il documento <u>Codifica Esportazione</u> .
esporta(\$filename, \$filename_err)	Rappresenta il motore dell'esportatore. Si occupa di prelevare i campioni, le analisi di questi, convertire i valori ed esportare i dati. I campioni e le analisi esportati correttamente saranno scritte nel file \$filename, mentre i campioni non esportati nel file \$filename_err.
esporta_campione_01(\$sample_number, \$date_completed, \$av_un_funzionale, \$av_codice_alimento, \$description, \$sampled_data, \$av_punto_prelievo, \$av_prelevatore, \$av_prodotto, \$av_distributore, \$av_conservazione, \$av_contenitore, \$date_reviewed)	Ritorna la stringa ottenuta dalla concatenazione di tutti i campi del campione da esportare (per il <i>record</i> di tipo 01).
esporta_campione_cont()	Ritorna la stringa ottenuta dalla concatenazione di tutti i campi del campione da esportare (per il <i>record</i> di continuazione).
esporta_analisi(\$cod_det, \$av_ministero, \$entry_qualifier, \$numerical_entry, \$in_spec)	Ritorna la stringa ottenuta dalla concatenazione dei campi dell'analisi del campione selezionato.
esporta_chiusura()	Ritorna la stringa ottenuta dalla concatenazione dei campi appartenenti alla chiusura. Il metodo stabilisce se deve ritornare una chiusura per un <i>record</i> di tipo 01 o di continuazione, in base alla variabile \$tipo_record.
esporta_riempimento(\$righe_analisi_report)	Ritorna una stringa contenente il riempimento del <i>record</i> . Dipende dalla variabile \$tipo_record.
setta_stringa_blank(\$len)	Ritorna una stringa di blank lunga quanto specificato dal parametro passato, che dovrà essere un intero non negativo.
setta_stringa_zeri(\$len)	Ritorna una stringa di zeri lunga quanto specificato dal parametro passato, che dovrà essere un intero non negativo.

<b>Nome</b>	<b>Descrizione</b>
cambia_server(\$server, \$login, \$password)	Cambia il server di connessione.
set_cod_pmp()	Setta il campo cod_pmp, secondo il nome server di connessione. In particolare, vengono settati: lab_tipo + cod_reg + cod_usl + prog_lab. (8B). Tale tabella è già fornita dal Ministero della Salute (Descrizioni PMP).
set_anno_camp(\$date_completed)	Setta il campo anno_camp secondo la codifica.
set_prog_camp(\$sample_number)	Setta il progressivo del campione. Se il sample_number ha più di 6 cifre, si tronca alla settima cifra a sinistra. Il riempimento deve essere fatto con zeri a sinistra.
set_esame_tipo(\$av_un_funzionale)	Setta il campo esame_tipo. L'esame può essere chimico (C), fisico (F) o biologico (B).
set_cod_alim_appart (\$av_codice_alimento)	Setta il campo cod_alim_appart
set_den_alim(\$description)	Setta il campo den_alim.
data_prel	Setta la data del prelievo del campione. Se questa è nulla, setta la data con zeri.
set_cod_localita_prel(\$av_punto_prelev o)	Setta il codice della regione, della provincia e del comune nel quale è stato effettuato il prelievo.
set_org_prelevatore(\$av_prelevatore)	Setta il campo org_prelevatore.
set_den_prod(\$av_prodotto)	Setta il campo den_prod.
set_reg_naz_prod()	Setta il campo reg_naz_prod.
set_den_dist(\$av_distributore)	Setta il campo den_dist.
set_mod_conserv(\$av_conservazione)	Setta il campo mod_conserv.
set_tipo_conten(\$av_contenitore)	Setta il campo tipo_conten.
set_note()	Setta il campo note.
set_data_fine_analisi(\$date_reviewed)	Setta il campo data_fine_analisi.
set_cod_det(\$av_ministero)	Setta il campo cod_det.
set_prog_det()	Setta il campo prog_det.
set_tecn_anal()	Setta il campo tecn_anal.
set_unit_mis()	Setta il campo unit_mis.
set_segno(\$entry_qualifier)	Setta il campo segno.
set_ris_sing_det(\$numerical_entry)	Setta il campo ris_sing_det.
set_esito_analisi(\$in_spec)	Setta il campo esito_analisi.

Tabella 3.2. I metodi della classe EsportatoreAlimenti.



### **3.3. Memorizzazione dei dati**

#### **Tipo, obiettivo e funzione della classe/interfaccia**

Rappresenta il database DB\_LIMS, ossia un database di una generica provincia dell'azienda ARPAV. Contiene i dati richiesti dal software *SEDAMS*. Questa componente è già realizzata.

#### **Relazioni d'uso di altre classi/interfacce**

Non utilizza il software *SEDAMS*.

#### **Interfacce e relazioni di uso da altre classi/interfacce**

E' utilizzato dalla classe EsportatoreAlimenti.

#### **Attività svolte e dati trattati**

Le attività svolte da questa componente sono:

- Ritornare le informazioni richieste alla logica applicativa.

I dati trattati sono:

- I dati specificati nel documento Analisi dei requisiti.

## 4. Codice sorgente

Il codice sorgente è in allegato al seguente documento. Comprende i file per l'interfaccia grafica, per l'esportazione dei dati, per la documentazione in linea e le immagini utilizzate. Di seguito si fornisce l'elenco dei file consegnati:

1. Cartella "src".
  - 1.1. File "manuale.html".
  - 1.1. File "sedams.html".
  - 1.2. Cartella "sedams".
    - 1.2.1. File "descrizione.php".
    - 1.2.2. File "errore.php".
    - 1.2.3. File "Esportatore.php".
    - 1.2.4. File "EsportatoreAlimenti.php".
    - 1.2.5. File "form.php".
    - 1.2.6. File "index.php".
    - 1.2.7. File "intestazione.php".
    - 1.2.8. File "sedams.php".
    - 1.2.9. File "verifica.php".
    - 1.2.10. File "stile.css".
    - 1.2.11. Cartella "images".
      - 1.2.11.1. File "nebulosa.jpg".
      - 1.2.11.2. File "stars.jpg".
      - 1.2.11.3. File "tellus\_luna.jpg".

Il software crea due cartelle denominate "export" e "not\_export" nello stesso path in cui è situato il file "sedams.html". In ognuna di queste due cartelle sarà creata un'ulteriore cartella "YYYYMMDD" dove YYYY, MM e DD rappresentano

rispettivamente l'anno, il mese ed il giorno in cui si esegue l'esportazione. All'interno di queste cartelle saranno successivamente posti il file di esportazione (nel path "export/YYYYMMDD/") ed il file dei campioni non esportati (nel path "not\_export/YYYYMMDD/"). Il nome di quest'ultimo file è del tipo:

sedams\_SiglaProvinciaVeneta\_err

## 5. Tracciamento della relazione componenti – requisiti

Nella tabella 5.1 è presente la matrice di tracciamento dei componenti con i requisiti specificati nel documento Analisi dei requisiti. Si può osservare che tutti i requisiti vengono tracciati nelle componenti definite in questo documento.

<b>MATRICE DI TRACCIAMENTO</b>	<b>GUI</b>	<b>Esportatore</b>	<b>DB_LIMS</b>
<b>RFA1</b>	x		
<b>RFA2</b>	x		
<b>RFA3</b>	x		
<b>RFA4</b>	x		
<b>RFA5</b>	x		
<b>RFA6</b>	x		
<b>RFA7</b>	x	x	
<b>RFA8</b>	x	x	
<b>RFA9</b>	x	x	
<b>RFA10</b>	x	x	
<b>RFA11</b>		x	x
<b>RFA12</b>		x	x
<b>RFA13</b>	x		
<b>RFB1</b>	x	x	
<b>RFB2</b>	x	x	
<b>RFB3</b>	x	x	
<b>RFB4</b>	x	x	
<b>RFC1</b>	x	x	
<b>RFC2</b>	x		

Tabella 5.1. Matrice di tracciamento componenti – requisiti.

## *Manuale utente*

**Nome del prodotto:**

Sistema di Esportazione Dati Alimenti per il Ministero della Salute  
(SEDAMS).

**Data:**

10/06/06

**Versione:**

1.0

**Stato del documento:**

Formale esterno

**Redazione:**

Piero Dalle Pezze

**Revisione:**

Piero Dalle Pezze    Padova, lì 10/06/06    Firma .....

**Approvazione:**

Paolo Zambotto    Padova, lì 10/06/06    Firma .....

**Lista di distribuzione:**

1. Paolo Zambotto
2. Piero Dalle Pezze

**Registro delle modifiche:**

v 1.0 10/06/06

Revisione formale del documento.

v 0.4 01/06/06

Scrittura capitolo 4.

v 0.3 31/05/06

Scrittura capitolo 2 e 3.

v 0.2 30/05/06

Scrittura capitolo 1.

v 0.1 29/05/06

Creazione del documento.

## Sommario:

Il presente documento rappresenta il manuale d'uso del software *SEDAMS* per l'utente.

## Indice:

1. Introduzione	152
1.1. Scopo del documento	152
1.2. Scopo del prodotto	152
1.3. Definizione dell'utente del prodotto	152
1.4. Come leggere il manuale	152
1.5. Documenti utili	152
1.6. Come riportare problemi e malfunzionamenti	153
2. Descrizione generale	154
3. Istruzioni per l'uso	156
3.1. Descrizione funzionale	156
3.1.1. Finestra di inserimento dati (form)	156
3.1.2. Finestra dell'evoluzione dell'esportazione	157
3.2. Azioni richieste / permesse	159
3.3. Errori probabili e cause possibili	159
4. Appendice	160
4.1. Messaggi di errore e loro significato	160
4.2. Glossario	161

## ***1. Introduzione***

Questo software è stato realizzato dallo stagista Dalle Pezze Piero per l'azienda ARPAV nell'anno 2006. Questo documento completa la documentazione del prodotto *SEDAMS* e costituisce inoltre una guida utile per il suo corretto utilizzo.

### ***1.1. Scopo del documento***

Lo scopo di questo documento è quello di fornire all'utente una descrizione del prodotto *SEDAMS*, del suo funzionamento e delle eccezioni che può sollevare.

### ***1.2. Scopo del prodotto***

Lo scopo del prodotto è quello di esportare dati nella codifica richiesta dal Ministero della Salute specificata nel documento *SA-99-D1006-015*.

### ***1.3. Definizione dell'utente del prodotto***

Questo prodotto software è rivolto al personale dell'azienda ARPAV incaricato all'esportazione dei campioni e delle analisi per il Ministero della Salute Italiana. E' necessario, quindi, che l'utente possieda un account, presso l'azienda ARPAV, con il permesso di esportare i dati dei campioni e delle loro analisi.

### ***1.4. Come leggere il manuale***

Il presente manuale può essere letto sia come aiuto nel caso in cui non si comprendano alcune caratteristiche del software *SEDAMS*, che per comprendere nel dettaglio la codifica del file di esportazione generato.

### ***1.5. Documenti utili***

Non sono indispensabili altri documenti per utilizzare correttamente il software. Nel caso si volesse leggere, comprendere e migliorare il codice sorgente, si



consiglia la documentazione ufficiale del linguaggio di programmazione PHP disponibile nel sito: <http://www.php.net/>.

### ***1.6. Come riportare problemi e malfunzionamenti***

Problemi e malfunzionamenti devono pervenire alla casella di posta elettronica [pzambotto@arpa.veneto.it](mailto:pzambotto@arpa.veneto.it) riportando come oggetto nella e-mail:

"Malfunzionamento software SEDAMS".

Le comunicazioni devono contenere le seguenti informazioni:

- ◆ Nome e Cognome mittente.
- ◆ Data riscontro del problema e/o malfunzionamento.
- ◆ Natura del problema e/o malfunzionamento.
- ◆ Descrizione dello scenario e delle condizioni sotto le quali si è verificato il problema e/o malfunzionamento.

L'azienda provvederà a gestire e risolvere il problema indicato, secondo le norme interne, per poi fornire un'eventuale versione del prodotto con le opportune correzioni.

## 2. Descrizione generale

*SEDAMS* è un acronimo per Sistema di Esportazione Dati Alimenti per il Ministero della Salute. Il suo compito è quello di generare un file di esportazione dei campioni autorizzati di tipo alimenti e delle loro analisi autorizzate. Il software provvede alla generazione di un file di esportazione per ogni provincia della regione Veneto come stipulato nel documento Analisi dei requisiti del software *SEDAMS*. Il personale addetto all'esportazione provvederà inseguito ad inviare il file unificato al Ministero della Salute Italiana come stabilito nel documento SA-99-D1006-015. Oltre al file di esportazione, il software *SEDAMS* genera anche il corrispondente file dei campioni che non sono stati esportati, in modo tale da poterli individuare e correggere all'interno del database dal personale ARPAV. Il file di esportazione è composto da *record* (righe) di 1000 caratteri. I *record* possono essere di due tipi: "01" o di continuazione ("02", "03", etc. ). Ogni *record* di tipo "01" individua un campione e le prime 16 sue analisi. Nel caso in cui un campione avesse oltre 16 analisi, allora viene generato un *record* di continuazione ("02") per quel campione nelle quali saranno scritte le successive 31 analisi. Nel caso in cui quest'ultimo non fosse ancora sufficiente, allora, sarà generato un ulteriore *record* di continuazione ("03") della stessa struttura del precedente e così via.

I due tipi di *record* hanno una struttura diversa. In generale, un *record* di tipo "01" è composto come di seguito specificato:

1. 463 caratteri sono dedicati ai dati anagrafici del campione (tipo del laboratorio, codice della regione, codice USL, il progressivo del laboratorio, il codice della struttura, etc.);
2. 31 caratteri x 16 analisi sono dedicati appunto per le prime 16 analisi del campione;
3. 40 caratteri per la chiusura.

Un *record* di tipo continuazione è invece composto come segue:

1. 22 caratteri sono dedicati ai dati anagrafici principali del campione in esame;
2. 31 caratteri x 31 analisi sono dedicati per le successive 31 analisi del campione;
3. 16 caratteri per la chiusura.

Si noti che i *record* sono di lunghezza fissa e quindi, nel caso in cui le analisi di un campione dovessero risultare minori di 16 (per il *record* di tipo "01") o minori di 31 (per un generico *record* di continuazione), si utilizzeranno ugualmente tutti i caratteri fino al completamento del millesimo.

Per una trattazione più completa, si rimanda al documento *SA-99-D1006-015*, nel quale si spiegano i campi dei campioni e delle analisi da esportare.

### ***3. Istruzioni per l'uso***

Il seguente capitolo illustra come utilizzare il software *SEDAMS* correttamente. In particolare, si descriveranno il funzionamento dettagliato del software e le azioni consentite all'utente e quelle necessarie.

#### ***3.1. Descrizione funzionale***

Il software è composto da due finestre principali: la finestra di inserimento dati e quella rappresentante l'evoluzione dell'esportazione. Di seguito, sono presentate nel dettaglio queste due finestre.

##### ***3.1.1. Finestra di inserimento dati (form)***

Per effettuare un'esportazione, l'utente deve compilare tutti i campi che costituiscono la form. Il significato e la modalità di compilazione è specificata di seguito.

1. Data Inizio: rappresenta la data a partire dalla quale si intende effettuare l'esportazione. Occorre specificare il giorno, il mese e l'anno posizionandosi con il mouse sulla freccia nera e selezionando il numero opportuno.
2. Data Fine: rappresenta la data entro la quale si intende effettuare l'esportazione. Occorre specificare il giorno, il mese e l'anno con le stesse modalità definite nel campo Data Inizio. I campi Data Inizio e Data Fine definiscono il periodo di esportazione.
3. Login: è la login dell'utente che intende effettuare l'esportazione. Tale login è definita internamente alla rete ARPAV
4. Password: è la password dell'utente che intende effettuare l'esportazione. Anch'essa è definita internamente alla rete ARPAV
5. Seleziona il server: occorre posizionare il mouse sulla freccia nera e

selezionare il server dal quale si intende effettuare l'esportazione. I nomi dei server sono così composti:

Database (DB) + Sigla della Provincia del Veneto + (LIMS).

6. Avvia Esportazione: è il bottone che permette di avviare l'esportazione.

### **3.1.2. Finestra dell'evoluzione dell'esportazione**

Dopo aver premuto il bottone "Avvia Esportazione", il software presenta in modo completo nel medesimo frame, i dettagli della sua esecuzione. Di seguito sono presentate le informazioni che il software *SEDAMS* rende disponibili.

1. Titolo: Ricorda all'utente quale server è stato selezionato.
2. Periodo di esportazione: Ricorda all'utente il periodo entro il quale il software sta effettuando l'esportazione.
3. Inizializzazione del path: Specifica le azioni del software nel filesystem. In particolare, il software deve testare l'esistenza delle cartelle export/ e not\_export/ path dove si trova la pagina sedams.html. All'interno di queste due cartelle, devono essere create due ulteriori cartelle con il nome YYMMDD (ossia anno, mese e giorno) correnti (es. 060515) che definiscono la data di esportazione. In questo modo, il software evita la sovrascrittura dei file se sono stati creati in due date distinte, dal momento che apparterranno a due cartelle con nome distinto. Il software, invece, sovrascriverà i file nel caso in cui si effettuasse più di un'esportazione nella stessa data e dallo stesso server. Se occorresse effettuare più di un'esportazione dallo stesso server nello stesso giorno, l'utente dovrà manualmente copiare il file prima di effettuare la successiva esportazione.
4. Stato della transazione: Definisce se il software è riuscito a stabilire una connessione al server. In seguito provvede a stampare, per ogni campione prelevato, l'esito della sua esportazione e quante analisi possiede. Se un campione non viene esportato, a causa di nessun analisi selezionata, significa che il campo AV\_CAMPO1 della tabella COMPONENT è nullo per

tutte le analisi di quel campione. Infine, dopo aver esportato tutti i campioni e le loro analisi per il periodo specificato, il software dichiara la chiusura della connessione.

5. Visualizzazione delle statistiche dell'esportazione: Dopo aver terminato l'esportazione, il software *SEDAMS* presenta le statistiche principali. In particolare vengono riportate le seguenti informazioni:

- Numero dei campioni che sono stati prelevati.
- Numero dei campioni che sono stati esportati.
- Numero dei campioni che sono stati prelevati ma non esportati.
- Numero delle analisi che sono state prelevate ed esportate.
- Numero medio di analisi per campione (prelevato).
- Numero medio di analisi per campione (esportato).
- Dimensione del file di esportazione (in KB). Il significato del punto quello impiegato nel sistema di misura internazionale.
- Dimensione del file dei campioni non esportati (in KB). Il significato del punto quello impiegato nel sistema di misura internazionale.
- Durata della transazione (in ore, minuti, secondi).
- Path assoluto del file di esportazione.
- Path assoluto del file dei campioni non esportati.

6. Files di esportazione: Dopo aver eseguito un'esportazione, il software *SEDAMS* presenta l'elenco dei file di esportazione finora generati. In particolare fornisce i collegamenti ipertestuali, in modo tale che l'utente può facilmente visualizzarne il contenuto.

7. Files dei campioni non esportati: Dopo aver eseguito un'esportazione, il software *SEDAMS* presenta l'elenco dei file dei campioni non esportati, finora generati. In particolare fornisce i collegamenti ipertestuali, in modo tale che l'utente può facilmente visualizzarne il contenuto.

Per ritornare nella finestra principale di inserimento dati, è sufficiente cliccare con il mouse nel collegamento ipertestuale “Parametri per l'esportazione” presente nel paragrafo Breve descrizione, riportato a sinistra.

### ***3.2. Azioni richieste / permesse***

Come specificato nel [paragrafo 3.1.1](#) di questo documento, tutti i parametri sono obbligatori per effettuare un'esportazione. Tuttavia, occorre precisare che il software è sensibile all'incoerenza delle date inserite. In particolare, l'utente deve specificare le date nel seguente modo:

1. I campi Data Inizio e Data Fine devono essere date corrette. Il software segnala un errore in caso di inserimento di una data non esistente.
2. Il campo Data Inizio deve precedere di almeno un giorno il campo Data Fine.

### ***3.3. Errori probabili e cause possibili***

Da quando il software ha superato il test di accettazione, presso l'azienda ARPAV, non sono stati rilevati errori né durante l'esportazione e né sul suo funzionamento.

## 4. Appendice

Di seguito, sono presentati rispettivamente i messaggi di errore che il software *SEDAMS* può sollevare durante la sua esecuzione, il loro rispettivo significato ed un breve glossario.

### 4.1. Messaggi di errore e loro significato

Nella tabella 4.1 sono riportati gli errori che il software *SEDAMS* può sollevare, le circostanze e la modalità di risoluzione.

<b>Messaggi di errore</b>		
<b>N°</b>	<b>Descrizione</b>	<b>Risoluzione</b>
<b>001</b>	Il campo <u>Data Inizio</u> non è una data corretta.	Controllare di aver inserito un giorno valido per quella data.
<b>002</b>	Il campo <u>Data Fine</u> non è una data corretta.	Controllare di aver inserito un giorno valido per quella data.
<b>003</b>	Il campo <u>Data Fine</u> precede il campo <u>Data Inizio</u> .	La data di inizio deve precedere la data di fine, altrimenti il periodo di esportazione non è consistente. Controllare il giorno, il mese e l'anno delle due date.
<b>004</b>	La cartella principale non può essere creata.	Controllare che il software <i>SEDAMS</i> abbia i permessi di scrittura nella sua cartella. Se non si dispongono dei privilegi, contattare l'amministratore di sistema.
<b>005</b>	La cartella secondaria non può essere creata.	Controllare che il software <i>SEDAMS</i> abbia i permessi di scrittura nelle cartelle export/ e not_export/. Se non si dispongono dei privilegi, contattare l'amministratore di sistema.
<b>006</b>	Il file di esportazione non può essere rimosso per la riscrittura.	Controllare che il software <i>SEDAMS</i> abbia i permessi di scrittura nelle cartelle export/YYYYMMDD/ e not_export/YYYYMMDD. Se non si dispongono dei privilegi, contattare l'amministratore di sistema.
<b>007</b>	Impossibile stabilire la connessione con il server selezionato.	Può essere dovuto a problemi di connessione o alla disconnessione del server al quale ci si intende connettere. Attendere o contattare l'amministrazione di sistema.
<b>008</b>	Impossibile aprire il file di esportazione per la scrittura.	Controllare che il software <i>SEDAMS</i> abbia i permessi di scrittura nelle cartelle



<b>Messaggi di errore</b>		
		export/YYYYMMDD/ e not_export/YYYYMMDD. Se non si dispongono dei privilegi, contattare l'amministratore di sistema.
<b>009</b>	Impossibile aprire il file dei campioni non esportati per la scrittura.	Controllare che il software <i>SEDAMS</i> abbia i permessi di scrittura nelle cartelle export/YYYYMMDD/ e not_export/YYYYMMDD. Se non si dispongono dei privilegi, contattare l'amministratore di sistema.

Tabella 4.1. Messaggi di errore che il software *SEDAMS* può sollevare.

## 4.2. Glossario

La tabella 4.2 costituisce un breve dizionario utente per il software *SEDAMS*. In essa sono contenuti tutti i termini che si presume l'utente non conosca. Nel documento, i termini che compaiono nel glossario, sono caratterizzati dal formato in corsivo.

<b>Dizionario utente</b>	
<b>Nome</b>	<b>Descrizione del nome</b>
<b><i>Form</i></b>	Detta anche Modulo. Nel linguaggio HTML, una <i>form</i> è un'area dedicata all'inserimento di alcuni parametri. In generale una <i>form</i> è costituita da alcuni campi che l'utente può o deve riempire e da un bottone di invio, con lo scopo di inviare i campi compilati.
<b><i>Frame</i></b>	Nel linguaggio HTML, i <i>frame</i> servono per separare la finestra in diverse parti, tante quante sono i <i>frame</i> . Di fatto, una pagina contenente <i>frame</i> , in realtà contiene delle pagine al suo interno.
<b><i>Path</i></b>	E' il cammino dove risiede un file. Può essere relativo o assoluto. Nel primo caso è relativo rispetto ad un altro file, nel secondo rispetto alla directory principale.
<b><i>Record</i></b>	Rappresenta una riga di 1000 caratteri nel file di esportazione. I <i>record</i> possono essere di due tipi: "01" o di continuazione ("02", "03", etc. ). Ogni <i>record</i> di tipo "01" individua un campione e le prime 16 sue analisi. Nel caso in cui un campione avesse oltre 16 analisi, allora viene generato un <i>record</i> di continuazione ("02") per quel campione nelle quali saranno scritte le successive 31 analisi. Nel caso in cui quest'ultimo non fosse ancora sufficiente, allora, sarà generato un ulteriore <i>record</i> di continuazione ("03") della stessa struttura del precedente e così via.
<b><i>SEDAMS</i></b>	E' un acronimo che indica questo software. In particolare significa: Sistema di Esportazione Dati Alimenti per il Ministero della Salute.
<b><i>Test di accettazione</i></b>	Detto anche test di collaudo. Serie di test dinamici eseguiti con lo scopo di accertare che i requisiti utente siano soddisfatti dal sistema.

Tabella 4.2. Dizionario utente.

## *Ringraziamenti*

Ringrazio il tutor aziendale Paolo Zambotto per la disponibilità e le attenzioni che mi ha concesso ed il Prof. Alessandro Sperduti per avermi costantemente seguito durante il periodo di stage e la realizzazione di questa tesi.

Vorrei anche ringraziare il mio collega di studi e compagno di stanza presso l'Agenzia ARPAV, Alberto Veronese per le ore piacevoli di lavoro.

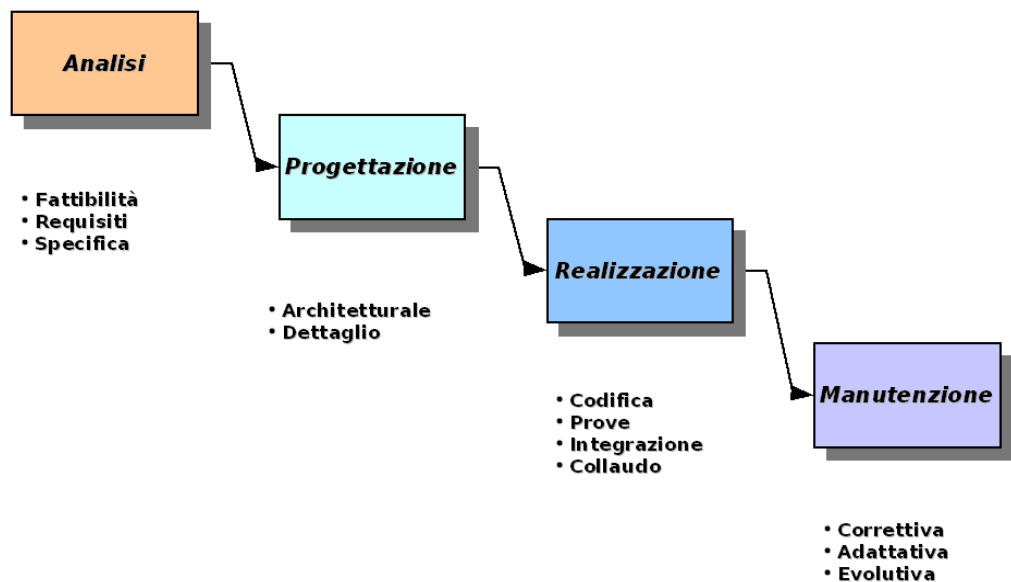
Infine, vorrei ringraziare la mia famiglia per avermi concesso di studiare presso quest'Università.

# Glossario

## A cascata

Modello di ciclo di vita in cui le fasi vengono svolte in modo sequenziale.

E' necessario stabilire con rigore, le pre-condizioni e le post-condizioni di ogni processo che si intende attuare. In questo modello, assume fondamentale importanza la documentazione. Per tale motivo si dice che è un modello *document-driven*.



Modello di ciclo di vita *waterfall*.

## Anomalia

Comportamento o caratteristica anomala di un prodotto intermedio. Ad esempio malfunzionamenti o deviazioni dalle norme di progetto di un prodotto intermedio.

## Attivo

Caratteristica di un modulo o di una componente. Significa che usa le funzionalità offerte da un altro modulo o componente.

### **Black box**

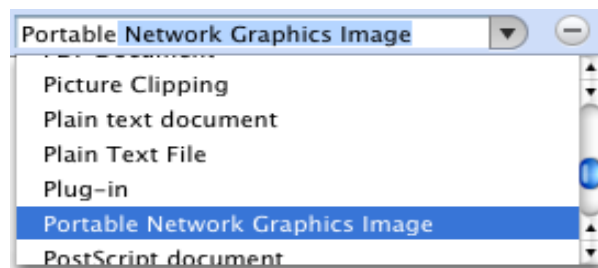
Modalità di esecuzione dei test dinamici. Consiste nel considerare l'unità come una scatola chiusa, studiando i suoi output senza analizzare il codice e i flussi di esecuzione.

### **Collaudo**

Serie di test dinamici eseguiti con lo scopo di accertare che i requisiti utente siano soddisfatti dal sistema.

### **Combobox**

Una combobox è una componente grafica che permette di selezionare un valore da un elenco. L'utente, cliccando con il mouse sulla freccia della combobox, apre il menù dei valori ed è in grado di selezionarne uno.



Esempio di combobox.

### **Componente**

Elemento costitutivo di un sistema software. Raggruppa funzionalità strettamente legate tra loro, e con una forte indipendenza dall'esterno.

### **CSS**

Acronimo di **Cascading Style Sheets**, definisce, mediante fogli di stile, la formattazione di pagine XHTML e HTML.

### **DBMS**

Acronimo di **Database Management System**. Un DBMS è costituito da

un insieme di programmi software che controllano l'organizzazione, la memorizzazione e il reperimento dei dati (campi, record e archivi) in un database. Normalmente, un DBMS è uno strumento per la creazione e la gestione efficiente di grandi quantità di dati che consente di conservarli in modo sicuro per lunghi periodi di tempo.

### **Design pattern**

Nell'ingegneria del software, un design pattern può essere definito "una soluzione progettuale generale a un problema ricorrente". Esso non è una libreria o un componente di software riusabile, quanto una descrizione o un modello da applicare per risolvere un problema che può presentarsi in diverse situazioni durante la progettazione e lo sviluppo del software.

I design pattern si classificano fundamentalmente in creazionali, strutturali, comportamentali, architetturali, metodologici.

### **Discrepanza**

Diversità delle caratteristiche di un prodotto intermedio rispetto alle caratteristiche attese. Può essere totale se non rispetta completamente le richieste o parziale se le rispetta solo in parte.

### **Errore**

Un errore (in inglese *error*) è qualsiasi cosa che non è conforme alla specifica di funzionamento del sistema. Un errore può causare uno o più fallimenti.

### **Fallimento**

Un fallimento (in inglese *failure*) è un evento per cui un sistema viola definitivamente le specifiche di funzionamento, interrompendo la disponibilità dei servizi che fornisce.

### **Form**

E' una parte di programma che permette l'inserimento dei dati da parte dell'utente. Normalmente una form si riferisce all'inserimento dati in una pagina web.

## **Frame**

Nel linguaggio HTML, i frame servono per separare la finestra in diverse parti, tante quante sono i frame. Di fatto, una pagina contenente frame, in realtà contiene delle pagine al suo interno.

## **Guasto**

Un guasto (in inglese *fault*) è un difetto (cioè la non conformità strutturale o algoritmica alla specifica) di un componente di un sistema. Un guasto può causare uno o più errori.

## **GUI**

Acronimo di **Graphical User Interface**. Sinonimo di interfaccia grafica.

## **Help**

Funzione richiamabile dall'utente attraverso l'interfaccia grafica con il compito di visualizzare la guida in linea.

## **HTML**

Acronimo di **Hyper Text Mark-Up Language**. E' un linguaggio usato per descrivere i documenti ipertestuali disponibili nel Web. Non è un linguaggio di programmazione, ma un linguaggio di markup, ossia descrive il contenuto, testuale e non, di una pagina web.

## **Inspection**

Metodo che prevede una lettura del codice focalizzata solo sulle zone più a rischio di errori.

## **Modulo**

Insieme di funzionalità affini che fanno parte una componente. Un modulo in genere è costituito da una classe o un'interfaccia (nel caso della progettazione OO).

## **Open Source**

(Termine inglese che significa sorgente aperto). Indica un software rilasciato con un tipo di licenza per la quale il codice sorgente è lasciato alla disponibilità di eventuali sviluppatori, in modo che con la collaborazione (in genere libera e spontanea) il prodotto finale possa

raggiungere una complessità maggiore di quanto potrebbe ottenere un singolo gruppo di programmazione. L'open source ha ovviamente tratto grande beneficio da internet.

### **Passivo**

Caratteristica di un modulo o di una componente. Significa che si limita ad offrire funzionalità che saranno usate da altri moduli o componenti.

### **Path**

E' il cammino dove risiede un file. Può essere relativo o assoluto. Nel primo caso è relativo rispetto ad un altro file, nel secondo rispetto alla directory principale.

### **PDCA**

Il ciclo PDCA (in inglese **Plan Do Check Act**) è una metodologia che garantisce una pianificazione iniziale, lo svolgimento dell'attività pianificata, il controllo e la sua correzione nel caso fossero riscontrate delle anomalie.

### **PHP**

**PHP** è un linguaggio di programmazione interpretato con licenza open source, utilizzato principalmente per la realizzazione di applicazioni web lato server e pagine web dinamiche. Da qui il suo nome: un acronimo ricorsivo che sta per **PHP: Hypertext Preprocessor**. PHP era in origine una raccolta di script CGI che permettevano una facile gestione delle pagine personali. Per questo motivo il significato originario dell'acronimo pare fosse **Personal Home Page**. Esso è un linguaggio facilmente estensibile, utilizzabile per la connessione a diversi database, supporta numerose tecnologie e prevede la possibilità di integrarsi nel codice HTML in modo da semplificare la realizzazione di pagine dinamiche. PHP riprende per molti versi simile al C; riprende però anche alcune caratteristiche sintattiche del Perl, per facilitare l'apprendimento a chi già conosce questo linguaggio. È un linguaggio debolmente tipato e, partire dalla versione 5, si può dire che supporta

seriamente la programmazione ad oggetti.

## **PL/SQL**

Si tratta del linguaggio SQL integrato con i costrutti della programmazione procedurale.

## **Prodotto intermedio**

Qualsiasi output prodotto dall'azienda (es. documenti, prodotto stesso).

## **Query**

Denota un'interrogazione ad un database, locale o remoto, in modo da ottenere dei dati.

## **Record**

Un record in questa tesi è una riga di 1000 caratteri nel file di esportazione. I record possono essere di due tipi: "01" o di continuazione ("02", "03", etc. ). Ogni record di tipo "01" individua un campione e le prime 16 sue analisi. Nel caso in cui un campione avesse oltre 16 analisi, allora viene generato un record di continuazione ("02") per quel campione nelle quali saranno scritte le successive 31 analisi. Nel caso in cui quest'ultimo non fosse ancora sufficiente, allora, sarà generato un ulteriore record di continuazione ("03") della stessa struttura del precedente e così via. I dettagli sulla struttura di tale record sono presenti nel documento ministeriale SA-99-D1006-015.

## **Requisiti di interfacciamento**

Un requisito di interfacciamento esplica come l'utente deve interfacciarsi al sistema o con una sua parte (definita dal contenuto del requisito).

## **Requisiti di qualità**

Un requisito di qualità è un requisito non funzionale che pone un vincolo alla progettazione o alla realizzazione del sistema. Per esempio, l'obbligo di utilizzare uno specifico linguaggio di programmazione rappresenta un classico requisito di qualità.

## **Requisiti funzionali**

Un requisito funzionale è un requisito che definisce indistintamente una



funzione del sistema. L'insieme dei requisiti funzionali dovrebbe rappresentare la descrizione del funzionamento del sistema.

### **Requisito desiderabile**

Un requisito è desiderabile se aumenta il valore del prodotto software, ma non è fondamentale.

### **Requisito obbligatorio**

Un requisito è obbligatorio se è necessario affinché il sistema funzioni correttamente (secondo anche le aspettative concordate con il cliente).

### **Requisito opzionale**

Un requisito è opzionale se non è necessario e può essere aggiunto in seguito senza perdita di benefici.

### **SEDAMS**

E' l'acronimo di Sistema di Esportazione Dati Alimenti per il Ministero della Salute. E' il nome del software che ho realizzato e che discuto in questa tesi.

### **SIRAV**

E' l'acronimo di Sistema Informativo Regionale Ambientale del Veneto.

### **SQL**

Acronimo di Structured Query Language. E' un linguaggio di programmazione strutturato con lo scopo di manipolare la struttura ed i dati di database relazionali. Fu adottato come standard per la prima volta dall'organizzazione [ANSI](#) (American National Standards Institute) nel 1986 e dall'organizzazione [ISO](#) (International Standard Organization) nel 1987.

### **Test di unità**

Serie di test dinamici eseguiti su una unità.

### **Test di integrazione**

Serie di test dinamici eseguiti durante l'assemblaggio di più unità, o durante l'assemblaggio di elementi che sono il risultato di precedenti integrazioni.

**Test di sistema**

Serie di test dinamici eseguiti con lo scopo di verificare il comportamento dinamico dell'intero sistema rispetto ai requisiti software.

**Test di accettazione**

Vedi collaudo.

**Test strutturale**

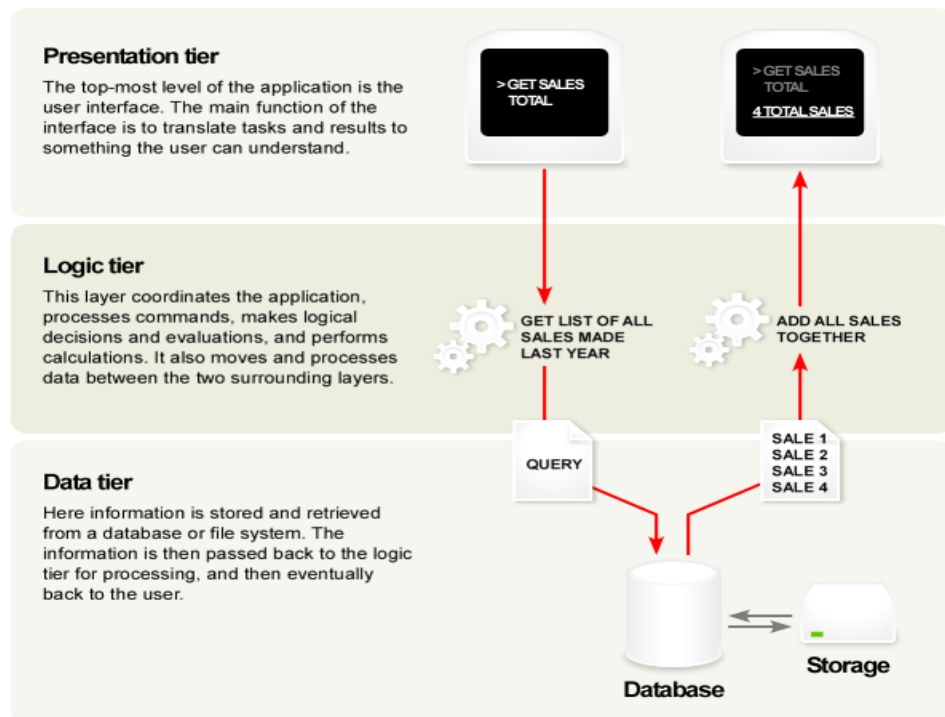
Test dinamico eseguito in modalità white box.

**Test funzionale**

Test dinamico eseguito in modalità black box.

**Three-tier**

Architettura client-server concepita su tre livelli applicativi. La maggior parte delle applicazioni web sono basate sul paradigma client/server a due livelli (ingl., two-tier). Diversamente, nell'architettura client-server three-tier sono tre i livelli, i quali possono essere ospitati su host distinti, anche con sistemi operativi eterogenei. Nell'architettura three-tier i tre strati applicativi sono costituiti ad esempio dal browser (lato client), dal server web con la web application e dal DBMS (entrambi lato server). Il browser effettua una richiesta al tier intermedio (logica di produzione) il quale accede, ed è l'unico a farlo, al DBMS. Per ultimo, il DBMS esegue l'operazione richiesta (query) su uno o più database, anche distribuiti. Nell'ingegneria del software, tale modello è un design pattern architetturale.



Il design pattern three-tier<sup>14</sup>.

## UML

Acronimo di **Unified Modeling Language**. E' un linguaggio di modellazione e specifica basato sul paradigma Object-Oriented. Il nucleo del linguaggio fu definito nel 1996 da Grady Booch, Jim Rumbaugh e Ivar Jacobson, sotto l'egida del gruppo [OMG](#), che tuttora gestisce lo standard di UML. Il linguaggio nacque con l'intento di unificare approcci precedenti (dovuti ai tre padri di UML e altri), raccogliendo le *pratiche migliori* (*best practice*) nel settore e definendo così uno standard industriale unificato.

## Walkthrough

Metodo di verifica mediante lettura del codice.

## White box

Modalità di esecuzione dei test dinamici. Consiste nel considerare l'unità come una scatola aperta, analizzare il codice e i flussi di esecuzione.

14. Immagine fornita da Wikipedia. (Cfr. [http://en.wikipedia.org/wiki/Main\\_Page/](http://en.wikipedia.org/wiki/Main_Page/))

## ***Bibliografia***

- Abran A. & Moore J.W. (2004). **Guide to the Software Engineering Body of Knowledge**. IEEE Computer Society. Software Engineering Coordinating Committee.
- **Documentazione ufficiale** del linguaggio PHP, reperibile nel sito <http://www.php.net/>
- Documento ministeriale **SA-99-D1006-015**.
- Loney K. (2004). **Oracle Database 10g: The complete Reference**. The McGraw-Hill Companies.
- Sommerville I. (2004). **Software Engineering**, 7th Ed. Addison Wesley.
- **Unified Modeling Language: Superstructure**. (v2.0, 2004), reperibile nel sito <http://www.omg.org/docs/formal/05-07-04.pdf>