

# Università di Padova

Corso di Ingegneria del Software

**Appalto concorso per la realizzazione di un  
modulo per la gestione di Curricula Vitae da  
integrarsi in un'applicazione esistente**

Mauro Michele  
versione del 31 Ottobre 2006

## > 1. Oggetto dell'appalto

Il presente capitolato concerne l'affidamento a fornitori terzi, della realizzazione di un modulo software prototipale per la gestione (inserimento, valutazione, ricerca, reportistica) di Curriculum Vitae di candidati e dei relativi metadati. Il modulo realizzato dovrà essere predisposto per la sua integrazione all'interno di un'applicazione esistente.

Il termine "VisioneHR-CV" denota il prodotto software in oggetto.

Il termine "committente" denota i proff. Vardanega e Conte, in qualità di delegati rappresentanti dell'azienda che ha commissionato il lavoro.

## > 2. Caratteristiche e requisiti tecnici minimi

### Premessa

Il progetto qui illustrato è intitolato:

*Modulo per la gestione di informazioni riguardanti Curriculum Vitae di persone da integrare in una applicazione esistente in architettura J2EE.*

Tale progetto, nel seguito denominato "progetto specifico", si colloca nel quadro di un progetto aziendale strategico più ampio, nel seguito denominato "progetto globale".

### Obiettivo del progetto globale

Il progetto specifico fa parte dello sviluppo attraverso l'aggiunta di nuove funzionalità del prodotto **VisioneHR** che il committente utilizza a supporto della propria attività consulenziale in ambito aziendale e di gestione delle risorse umane.

Il prodotto completo gestisce in maniera unificata vari domini informativi che descrivono diversi aspetti del rapporto fra le persone e l'azienda presso cui lavorano, con particolare focalizzazione sul percorso formativo delle persone e sulla valutazione delle loro competenze, delle loro prestazioni e del clima aziendale in cui si trovano.

Il prodotto è strutturato come un'applicazione *Web J2EE* fornita da un *application server* e utilizzata attraverso un *browser*. I dati sono memorizzati in un database secondo un modello dati che ha dimostrato finora ottime caratteristiche di flessibilità nell'accogliere nuovi requisiti e nuove funzioni.

L'architettura secondo la quale il prodotto è stato disegnato consente con relativa facilità di cambiare sia l'*application server* ospite che la base dati di supporto, permettendo di venire incontro sia ai clienti che non avessero una forte struttura IT

interna (implementando la soluzione con una coppia *application server* – base dati gratuita) sia a quelli che già dispongono di un *data center* organizzato (andando quindi ad appoggiarsi ai servizi esistenti, senza richiedere uno specifico *vendor*).

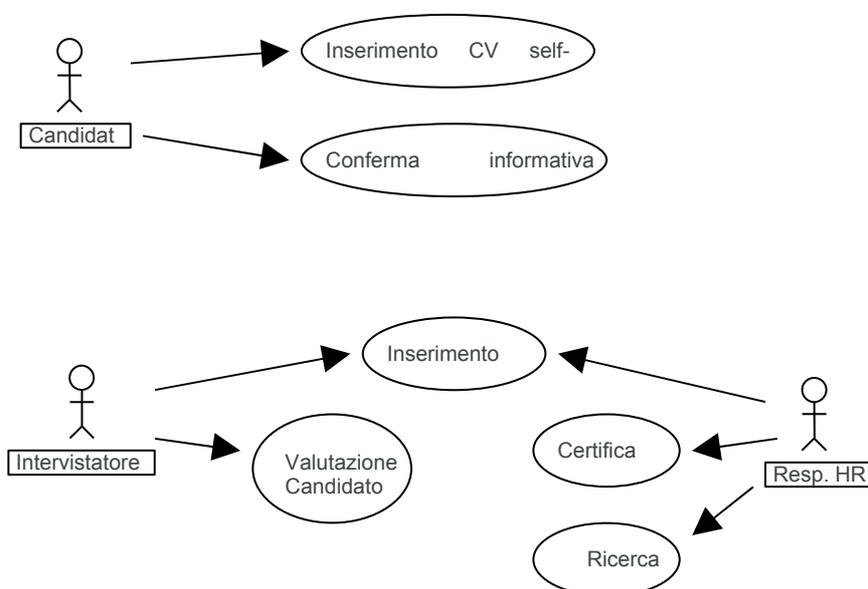
### Stato dell'arte

Attualmente, il prodotto è già stato installato ed è in uso presso diversi clienti, con personalizzazioni che differenziano ogni installazione a seconda delle specifiche necessità di ciascun cliente.

Il fornitore vincitore dell'appalto concorderà con il committente i requisiti del modulo da sviluppare, sia in termini funzionali che in termini di integrazione con l'esistente. Verranno condivise con il committente tutte le informazioni necessarie a far sì che il modello dati sia contemporaneamente adeguato alle funzioni richieste e in armonia con la situazione esistente.

### Obiettivo del progetto specifico

Partendo dagli obiettivi del progetto globale come precedentemente illustrati ed alla luce dello stato dell'arte attuale, l'obiettivo primario del progetto specifico essenzialmente consiste nella gestione dei seguenti casi d'uso:



Casi d'uso da parte dell'attore 'Candidato':

- *Inserimento CV self-service*: il candidato, in autonomia, compila il CV con i propri dati. Ciò tipicamente potrebbe avvenire presso un terminale negli uffici dell'azienda, ma la funzionalità potrebbe essere esposta anche all'interno del sito *Internet* rivolto al pubblico.
- *Conferma informativa Privacy*: il candidato riceve un *link* personalizzato ed unico per lui che gli consente di prendere visione dell'informativa dettagliata riguardante la gestione dei suoi dati personali ed esprimere in maniera esplicita il suo consenso.

Casi d'uso da parte dell'attore 'Responsabile HR':

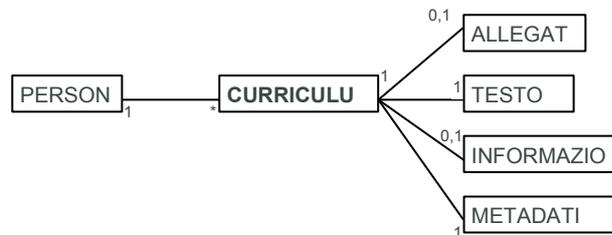
- *Inserimento CV*: il responsabile dell'ufficio HR inserisce nel sistema un CV ricevuto da un candidato (via posta elettronica, in forma cartacea, per posta, ecc.) compilando le informazioni che ritiene utili e/o allegando un testo o un *file*.
- *Certifica CV*: il responsabile HR consulta la lista dei CV inseriti in modalità *self-service* o per i quali è necessario l'ottenimento del consenso relativo alla *Privacy*. Per i primi, completa l'inserimento a sistema con eventuali dati accessori; per i secondi, comanda l'invio alle persone interessate delle istruzioni e del *link* che consentono a queste ultime di perfezionare il loro consenso.
- *Ricerca CV*: il responsabile HR ricerca all'interno del sistema curriculum che rispondano a criteri specificati, per es. competenze richieste, età del candidato, eccetera.

Casi d'uso da parte dell'attore 'Intervistatore':

- *Inserimento CV*: l'intervistatore inserisce nel sistema un CV ricevuto da un candidato (via posta elettronica, in forma cartacea, per posta, ecc.) compilando le informazioni che ritiene utili e/o allegando un testo o un file.
- *Valutazione Candidato*: l'intervistatore inserisce una valutazione del candidato, secondo criteri predefiniti a sistema, a corredo del colloquio o conseguentemente all'esame del CV. Questa funzionalità potrebbe integrarsi con il sistema di gestione di valutazioni già presente nell'applicazione principale, condividendone come minimo il modello dati.

## Modello dati

A supporto della conservazione dei dati, è possibile ipotizzare un modello dati strutturato nel modo seguente:

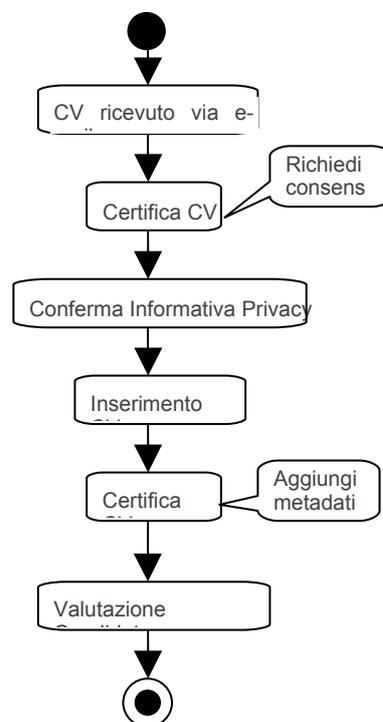


La principale entità gestita dal progetto specifica sarà il Curriculum, che verrà riferito ad una entità Persona già esistente nel modello dati dell'applicazione principale. Una persona può avere più curriculum legati ad essa. Il curriculum potrà avere associato un Allegato, inteso come file binario (pdf o altro formato applicativo), ed un Testo anche lungo (che potrebbe per esempio contenere un curriculum inviato come mail in formato testo). Al curriculum corrisponderanno un insieme di Informazioni strutturate in maniera assolutamente arbitraria (per es. competenze raccolte in vario modo, abilità specifiche, lingue conosciute). In fase di definizione dei requisiti potrà essere deciso l'eventuale collegamento ad entità già esistenti nell'applicazione principale. Infine, il curriculum dispone di un insieme di Metadati, ovvero attributi quali la data di inserimento, la modalità di inserimento (*self-service*, imputato internamente, ecc.), il canale di provenienza (visita personale, posta ordinaria, email) e così via.

### Gestione Processi

Il processo di ricezione e valutazione di curriculum può attraversare fasi diverse e attività eseguite da ruoli differenti, a seconda dal modo con cui viene ricevuto e delle specifiche esigenze del cliente. La gestione di questo processo deve essere affidata a un opportuno motore di **workflow** che sia configurabile in modo rapido e flessibile,

per potersi adattare alle diverse situazioni organizzative. Per fare un esempio, il *workflow* di gestione di un CV ricevuto via *e-mail* potrebbe essere il seguente:



*Alla ricezione di un CV via e-mail, il destinatario richiede il consenso al mittente della mail. Il candidato esprime il consenso come descritto nel caso d'uso, dopodiché la persona che ha ricevuto la mail può mettere nel sistema il contenuto del CV. A questo punto, un responsabile dovrà aggiungere i metadati necessari per la certificazione del CV (per es. specifica dell'origine del CV, posizione richiesta, eventuali riferimenti ad annunci fatti, ecc.). Una volta che il CV è certificato, può*

*passare ad un intervistatore che valuterà il candidato esaminando quanto ricevuto ovvero contattando il candidato per un colloquio.*

Alcune situazioni potrebbero richiedere, al posto di una struttura lineare come quella esemplificata, delle ramificazioni a seconda, per esempio, della posizione per la quale viene inviato il CV: alcune posizioni potrebbero richiedere un tipo di valutazione, altre potrebbero non richiedere ulteriori approfondimenti. Oppure la fase di valutazione potrebbe essere in realtà composta di più attività, ovvero di più valutazioni fatte da diverse persone sui vari aspetti del CV.

Perché il sistema sia adattabile ai diversi requisiti dei clienti, la parte di gestione del *workflow* deve esprimere sufficiente generalità. Di qui la necessità di scegliere un motore di *workflow* generico e corredarlo del codice necessario a realizzare i casi d'uso, che poi verranno richiamati all'interno dell'esecuzione dei processi. Opportune *worklist*, integrate nell'interfaccia utente dell'applicazione, presenteranno ai vari utenti le attività che sono incaricati a svolgere.

Una corretta integrazione di un motore di *workflow* può inoltre aprire interessanti prospettive di estensione anche al resto del progetto globale.

## Ambiente di sviluppo

Il modulo oggetto del progetto specifico, per garantire una corretta integrazione con l'applicazione complessiva, dovrà dividerne l'architettura fondamentale, che si può sintetizzare nei seguenti punti:

- Linguaggio di programmazione Java versione 1.4.
- **Framework web Struts** 1 per la parte *controller*.
- **Java Server Pages** (JSP) per la parte *view*, con l'uso di javascript ove necessario (ma non di tecniche AJAX), e senza l'utilizzo di codice Java *embedded* se non assolutamente indispensabile.
- **Hibernate** versione 3.1 per la parte *model*, come *Object Relational Mapping* fra gli oggetti Java e la base dati relazionale.
- **Librerie JasperReports** per la produzione di *report* in formato PDF e Microsoft Excel.
- **Librerie JFreeCharts** per la produzione di grafici.
- Uso di librerie e componenti *Open Source* per evitare dipendenze da specifici *vendor* o moduli *software* di terze parti.

Queste caratteristiche sono state selezionate allo scopo di mantenere la massima compatibilità possibile rispetto ad ogni configurazione di *browser*, *application server* e base dati riscontrabile presso i clienti, e per questo motivo non necessariamente rispecchiano lo stato più avanzato della tecnologia, ma si fermano alle implementazioni ed alle strutture più stabili e consolidate di ciascuna parte del paradigma MVC.

## > 3. Documentazione

Con la consegna del modulo *software*, il fornitore assumerà anche l'onere di fornire la documentazione tecnica idonea all'integrazione del progetto specifico all'interno dell'applicazione esistente.

La documentazione è costituita, come minimo:

- dal codice sorgente del progetto specifico, commentato e comprensivo di tutti i file di configurazione necessari (configurazione di Struts, di Hibernate, del motore di *workflow*, eventuale web.xml per servlet specifiche)
- dal codice sorgente delle classi di *test* e dai dati necessari per l'esecuzione dei *test* stessi
- da un documento d'insieme che descrive le singole parti del progetto (azioni Struts, classi del modello, classi di supporto, pagine jsp, servlet, eccetera) e il modo in cui collaborano per realizzare le funzionalità richieste
- da un documento specifico che dettaglia le scelte operate nell'interfacciamento con il motore di *workflow*, le assunzioni e le scelte operate, e le opzioni di configurazione disponibili

## > 4. Rinvio

Per tutto quanto non previsto nel presente capitolato, sono applicabili le vigenti disposizioni di legge.