Integrazione GDA - DOC/ER

Analisi degli interventi

di AXIA Studio

Preparato per: Aleven di Mirco Attocchi

Preparato da: Tiziano Lattisi - AXIA Studio S.r.l.

Data: 8 agosto 2016

Stato documento: v 1.1

INTRODUZIONE

Gedoc DOC/ER

Il modello documentale GeDoc è un middle layer che espone una serie di servizi per la gestione documentale attraverso delle interfacce standard, e che si interfaccia con servizi server qualificati per fornirli.

La sua implementazione DOC/ER[1] è fornita già correlata con alcuni servizi di sistema (identificazione, autorizzazione, auditing) nonché alcuni servizi aggiuntivi, come il servizio documentale e quello di invio in conservazione verso Sacer, il servizio di conservazione del Polo Archivistico dell'Emilia Romagna.

Quest'ultima funzionalità permetterebbe di evitare gli elevati investimenti necessari per aderire direttamente a Sacer.

La scelta del modello

Seguendo questo modello, un software di protocollo dovrebbe esporre all'implementazione DOC/ER un servizio di protocollazione seguendo delle specifiche interfacce Java, divenendo quindi "protocollo qualificato". In questo modo anche applicativi verticali potrebbero integrarsi nel sistema di gestione documentale attraverso apposite chiamate al middle-layer.

L'ipotesi centrale di questo progetto è quella di utilizzare DOC/ER, almeno in questa prima fase, non come middle-layer per la gestione del processo documentale, ma come servizio di archiviazione al termine del processo di protocollazione.

Come verrà spiegato meglio più avanti, questo approccio è pensato per essere una soluzione transitoria. Il passo successivo previsto è l'implementazione dei servizi di protocollazione, fascicolazione, e invio PEC[2].

Sposare il modello DOC/ER può portare a grandi benefici, specialmente se si desidera mantenere lo sviluppo di applicativi in casa. Di contro è un matrimonio che porta con se alcuni vincoli, sia di architettura che funzionali. Pertanto è necessario valutare coscientemente la scelta.

L'architettura

Le specifiche DOC/ER prevedono web service Java che comunicano attraverso il protocollo SOAP/HTTP.

Per ridurre le problematiche di disomogeneità di architettura, parte della soluzione sarà sviluppata web-based, limitando il lavoro sull'attuale client ad un'interfaccia con un numero limitato di chiamate.

---

[1] Da questo punto in poi ci riferiremo per semplicità unicamente a DOC/ER intendendo indistintamente il modello GeDoc e la sua implementazione.

[2] I servizi devono essere implementati in java seguendo le interfacce IProviderProtocollazione.java, IProviderFascicolazione.java, IProviderInvioPEC.java presenti come allegati alle specifiche tecniche.

In generale ormai da anni la tendenza è quella di integrare servizi tramite web service, di conseguenza ogni scelta che punti in quella direzione va automaticamente a favorire la longevità del progetto[3].

Scopo del documento

Lo scopo di questo documento è individuare i metodi e i servizi DOC/ER necessari per l'integrazione in GDA, nonché dare le principali indicazioni sul loro utilizzo.

Il fine ultimo è quello di permettere la costruzione di un capitolato e la stima di costi e tempi di integrazione.

In coda al documento una proposta di dettaglio di fornitura da utilizzare per un ordine diretto.

RIEPILOGO DI PROGETTO

Introduzione

Il lavoro di integrazione prevede che siano già presenti e operative alcune componenti. In questo capitolo distinguiamo quindi le componenti che sono date per "preesistenti", e quali sono invece oggetto di questa relazione e della fornitura.

Prerequisiti

Vengono dati per scontati questi requisiti:

- il sistema DOC/ER installato e correttamente configurato (con integrazione verso Sacer);
- la sincronizzazione degli utenti tra Postgres e DOC/ER[4];
- il titolario sincronizzato tra GDA e DOC/ER[5].

Sintesi interventi

Gli interventi possono essere ricondotti a tre punti:

- la gestione delle unità documentali in DOC/ER come spazi documenti collegati ai protocolli di GDA (capitolo "versamento" e "versionamento");

[3] Basti pensare alla possibilità di distribuire una parte (accesso ai documenti) dell'applicazione via browser o in mobilità.

[4] GDA verifica utente e password su Postgres, quindi sarà necessario che l'attuale autenticazione ntml di Postgres sia sostituita con un autenticazione sul medesimo server LDAP su cui può fare le query DOC/ER, o sistema che alternativamente garantisca la sincronizzazione delle credenziali.

[5] Le modifiche al titolarlo sono sporadiche, e possono quindi essere gestite con iniziale importazione massiva, e successivi aggiornamenti a mano. In caso di protocollazione (protocollaDocument) senza fascicolazione e classificazione questa sincronizzazione potrebbe essere opzionale.

- la funzionalità di invio di tali unità documentali in archivio di deposito affinché DOC/ER possa mandarle in conservazione tramite Sacer[6] (capitolo “archiviazione”);
- le funzionalità di sincronizzazione di alcune componenti GDA con le relative in DOC/ER (“sincronizzazione GDA DOC/ER”).

VERSAMENTO

Introduzione

GDA archivia in modo diretto sull’attuale ECM (Alfresco) i documenti relativi al protocollo. Per ognuno di questi c’è una folder sull’ECM costruita con un path recante riferimenti della segnatura del protocollo (anno, mese, giorno, numero).

Il middle layer DOC/ER, diversamente, prevede che dei documenti che fanno parte della medesima unità documentale siano collegati mediante la definizione di relazioni. In altre parole, i documenti associati ad un particolare contesto sono composti da un documento, e da tutti i suoi related.

Il protocollo GDA dovrà tenere un riferimento che permetta di accedere a tutti i documenti di questa unita documentaria. Tenere come riferimento il docId di uno dei documenti obbligherebbe alla gestione di un cambio di id di riferimento, qualora si decidesse di cancellare tale documento “radice”.

Per tale motivo si utilizzerà un oggetto Folder di DOC/ER per contenere tutti i documenti related dell’unità documentaria. L’id di questa folder sarà il riferimento mantenuto da GDA[7].

Per poter quindi inserire un documento (file) in DOC/ER nel contesto di un protocollo GDA è necessario creare una cartella, inserire un primo documento, e associargli come related i successivi (createDocument). Il primo documento, come detto, potrebbe anche essere cancellato; in questo caso i nuovi documenti inseriti è sufficiente vengano messi in relazione esplicita con uno qualunque dei documenti superstiti, lasciando al sistema la generazione delle restanti relazioni implicite.

DOC/ER gestisce dei profili di accesso ai documenti (setACLDocument), che dovranno rispecchiare quanto configurato in GDA (sportello e attribuzioni)[8].

createFolder

Il metodo createFolder crea una cartella in DOC/ER, restituisce un identificativo (folderId), e prende come parametri informazioni relative a nome, proprietario, e cartella padre.

---

[6] L’interfaccia DOC/ER verso PARER non è oggetto di questo progetto.

[7] La relazione tra documenti della stessa unità documentaria è sempre un grafo completo, poiché vengono automaticamente generate anche le relazioni non esplicitamente definite.

[8] GDA prevede specifiche regole di accesso lettura/scrittura per gli uffici (gruppi) definiti come sportello, come attribuzione principale, o secondaria; queste regole dovranno essere rispettate nelle definizioni delle ACL.

Metadato	Obbligatorio	Descrizione
FOLDER_NAME	sì	il nome della cartella
FOLDER_OWNER	sì	utente proprietario della cartella

Attraverso questo identificativo, mantenuto in GDA, sarà possibile accedere a tutti i documenti contenuti.

createDocument

Il metodo createDocument inserisce il documento con alcuni metadati, tra cui:

Metadato	Obbligatorio	Descrizione
TYPE_ID	sì	
DOCNAME	sì	il nome del documento comprensivo di estensione del file allegato
ABSTRACT	no	
TIPO_COMPONENTE	no	PRINCIPALE/ALLEGATO/ANNOTAZIONE/ANNESSO

Il metodo restituisce uno String contenente l'id del documento appena creato.

addToFolderDocuments

Il metodo addToFolderDocuments permette di inserire documenti in una cartella. Prende come parametro l'id della cartella e una lista di String con gli identificativi da includere, ovvero i docId restituiti dalle chiamate createDocument.

removeFromFolderDocuments

Simmetricamente il metodo removeFromFolderDocument rimuove i documenti inseriti in una cartella. Nel nostro caso questo metodo potrebbe non servire, in quanto l'unica situazione in cui ha senso la rimozione di un documento da una cartella è la sua cancellazione, presa in esame nel metodo seguente.

deleteDocument

Il metodo deleteDocument prende come unico parametro l'docId del documento da cancellare.

setACLDocument

Il metodo setACLDocument prende come parametro l'iddic di un documento, e una lista di nodi ACL. Limitatamente al nostro utilizzo, un nodo è composto da un identificativo dell'ufficio (GROUP_ID) e una permission.

Value	Access
2	Read only: lettura del profilo di documento e del file

Value	Access
1	Normal: aggiunge permessi di scrittura su profilo e file (ma non cancellazione)
0	Full: anche cancellazione del file e gestione delle ACL

addRelated

Il metodo addRelated prende come parametri l'docId di un documento, e una lista di docId di documenti che vogliamo collegare.

searchFolders

Il metodo searchFolders permette di eseguire ricerche tra le cartelle in DOC/ER. Prende come parametro una lista di criteri ma nel nostro contesto verrà utilizzato solo per l'estrazione della cartella associata la protocollo GDA tramite FOLDER_ID.

getFolderDocuments

Il metodo getFolderDocuments prende come parametro solo un folderd, e restituisce una lista con i docId dei documenti contenuti.

Questo metodo, in combinazione con il precedente searchFolders[9], permette di generare la lista dei documenti di un protocollo GDA.

Implementazione

La UI aperta da una registrazione di protocollo presenterà inizialmente una lista vuota di documenti. Se ancora non esiste un folderId associato, allora verrà invocata la creazione della cartella (createFolder).

A questo punto sarà possibile inserire documenti (createDocument e addToFolderDocuments), che dovranno essere messi in relazione con almeno uno dei documenti già esistenti nella cartella (addRelated)[10].

Contestualmente ad ogni creazione di documento sarà necessario applicare le corrette politiche di accesso (setACLDocument) per gli uffici sportello e attribuzioni[11].

Alle successive riaperture della UI, poiché è presente un folderId[12], la lista di file sarà popolata con i documenti contenuti (searchFolders e getFolderDocuments).

Note:

---

[9] In realtà in alcune circostanze potrebbe essere sufficiente l'utilizzo di getFolderDocuments senza serchFolders.

[10] Da valutare l'opportunità di generare la rete dei related solo l'istante prima della protocollazione. Questa decisione dipende dalle considerazioni che si faranno riguardo all'evoluzione dell'architettura.

[11] Dovrà essere deciso l'access level (0, 1, 2) per sportello, attribuzione principale, e altre attribuzioni.

[12] Il meccanismo affinché ciò sia possibile potrebbe essere il seguente: alla prima apertura il client GDA passa l'id del protocollo, e la procedura incaricata di eseguire il createFolder si occupa di aggiornare il record di protocollo con folderId.

- ad ogni modifica di sportello o attribuzione dovrà essere eseguita la setACLDocument su tutti i documenti della cartella;
- in questa fase non viene ancora valorizzato il metadato TIPO_COMPONENTE, poiché non è possibile inserire documenti con tipo ALLEGATO successivamente al documento (unico) con tipo PRINCIPALE.

VERSIONAMENTO

Introduzione

DOC/ER prevede due tipi di versionamento: standard, per la gestione del singolo documento, e avanzato, per il versionamento dell'intera unità documentale (quindi metadati e struttura di allegati, annotazione, etc).

Per i nostri scopi è sufficiente implementare la standard, in quanto le differenti versioni si riferiscono al singolo documento, e non sono previsti cambi di tipo MIME.

addNewVersion

Il metodo addNewVersion prende come parametro il docId del documento da versionare, e un base64Binary con il contenuto della nuova versione. Viene restituito uno String con il numero della nuova revisione.

getVersions

Il metodo getVersions prende come parametro il docId di un documento, e restituisce la lista dei numeri di versione presenti. Questi numeri possono essere utilizzati dal metodo downloadVersion.

downloadVersion

Il metodo downloadVersion prende come parametro un docId e un numero di versione, e restituisce il documento (file) nella revisione richiesta.

Implementazione

La UI con la lista dei documenti, di protocollo o fascicolo, deve permettere la selezione di un documento per l'upload di una nuova versione (addNewVersion). Su richiesta deve esporre l'elenco delle revisioni del documento (getVersions) con la possibilità di scaricarne una specifica (downloadVersion).

Note:

- le funzionalità di versionamento sono disponibili solo per i documenti con STATO_ARCHIVISTICO=0, di conseguenza sono disabilitate a partire dalla chiamata al metodo protocollaDocumento[13]

[13] Poiché questa regola collide con la politica di utilizzo di GDA, si prevede di eseguire la chiamata protocollaDocumento solo all'istante della mandata in conservazione.

PROTOCOLLAZIONE E ARCHIVIAZIONE

Introduzione

Con i metodi e le procedure descritte nel capitolo precedente otteniamo la possibilità di poter gestire dei gruppi di documenti, associati tramite un folderId (una cartella in DOC/ER) ad un protocollo in GDA.

I documenti inseriti in questo modo dovranno essere protocollati in DOC/ER prima di essere mandati in conservazione. Questo momento è stato individuato corrispondente al momento in cui in GDA viene eseguita l'operazione di "consolida" del protocollo, che implica anche nell'implementazione attuale diretta su ECM l'immodificabilità dei documenti.

Sarà quindi utilizzato il metodo di protocollazione DOC/ER (protocollaDocumento).

protocollaDocumento

Il metodo protocollaDocumento prende come primo parametro il docId del documento da protocollare, e come parametro successivo i metadati, tra cui:

Metadato	Obbligatorio	Descrizione
NUM_PG	sì	numero di registrazione del protocollo
ANNO_PG	no	anno di registrazione (se omesso è calcolato da DATA_PG)
DATA_PG	si	la data di protocollazione
OGGETTO_PG		oggetto del protocollo
REGISTRO_PG		registro del protocollo
TIPO_PROTOCOLLAZIONE		E (entrata), U (uscita), I (interno), ND (non definito)

Note:

- il documento da protocollare deve essere quello che diverrà principale;
- ai documenti related viene valorizzato il campo DOCNUM_PRINC con il docId del principale (vedi più avanti registrazione);
- il documento passa in stato archivistico 3 (Protocollato).

classificaDocumento

Il metodo classificaDocumento prende come parametro un docId e la classifica principale da assegnare (deve essere presente nel titolario sincronizzato). Classifica il documento corrispondente e tutti i suoi related.

archiviaDocumento

Il metodo archiviaDocumento manda in archivio un documento e i suoi related. Prende come parametro un docId, nel nostro caso sarà quello del documento selezionato in UI.

I documenti così archiviati passano in STATO_ARCHIVISTICO 6 (Archiviato).

Implementazione

Dalla UI con la lista dei documenti sarà possibile richiedere protocollazione e archiviazione, solo se il protocollo in GDA è in stato consolidato[14]. L'esecuzione del metodo protocollaDocumento imposta per il documento selezionato il metadato TIPO_COMPONENTE a PRINCIPALE, mentre per tutti i related verrà impostato ad ALLEGATO.

Da questo stato archivistico è possibile classificare il protocollo (classificaDocumento) richiedere l'archiviazione (archiviaDocumento) rimuovendo quindi i documenti dall'archivio corrente.

Note:

- la registrazione di protocollo su DOC/ER è differita rispetto alla registrazione in GDA, poiché è funzionale unicamente all'archiviazione, eseguita contestualmente[15];
- da questo momento in poi non sarà più possibile aggiungere, cancellare o versionare documenti[16].

SINCRONIZZAZIONE GDA DOC/ER

Utenti

Come detto poco sopra, la sincronizzazione degli utenti (credenziali di autenticazione) viene data per preesistente.

Ciò significa che un nuovo utente generato in GDA sarà attivo, ovvero sarà possibile autenticarsi con le sue credenziali, solo se presente anche in DOC/ER.

Anche l'appartenenza dell'utente ad uno o più uffici sarà possibile solo se l'utente e l'ufficio sono già presenti in DOC/ER, con il medesimo USER_ID e GROUP_ID.

Uffici

All'inserimento di un nuovo ufficio in GDA dovrà corrispondere all'inserimento del corrispondente gruppo in DOC/ER.

---

[14] La funzione potrebber essere richiamabile anche da client GDA, all'interno della procedura di consolida. In questo caso sarà però necessario predisporre un meccanismo per individuare l'id del documento che diverrà documento principale.

[15] Questo approccio serve a eliminare l'esigenza di mantenere sincronizzati dati di protocollazione tra GDA e DOC/ER, prima dell'invio in archivio.

[16] E' un vincolo dato dal fatto che sono operazioni permesse solo per documenti con STATO_ARCHIVISTICO = 0.

Ciò significa che al momento del salvataggio dovrà essere invocato su DOC/ER il metodo createGroup[17].

Gestione associazioni utente-ufficio e autenticazione

La maschera di gestione utenti non sarà modificata, ma saranno ritenuti validi solo gli utenti che hanno un identificativo riconosciuto all'interno del server LDAP. Se non presenti in LDAP, di fatto non potranno autenticarsi. In mancanza di un'implementazione LDAP, potrà essere allestito un sistema di autenticazione su DEC/ER diverso[18].

La creazione e la modifica di un nuovo ufficio comporterà l'invocazione dei metodi rispettivi:

- createGroup (GROUP_ID, GROUP_NAME, PARENT_GROUP_ID)
- updateGroup (GROUP_ID, GROUP_NAME, PARENT_GROUP_ID)

L'aggiunta e la rimozione di un ufficio (con flag visualizza) associato ad un utente comporterà l'invocazione sincrona dei metodi DOC/ER:

- setGroupsOfUser (groups[])
- updateGroupsOfUser (groupsToAdd[], groupsToRemove[])

VARIANTI

Introduzione

L'ipotesi di progetto esposta si presta a varianti che potrebbero essere prese in considerazione. In questo capitolo verranno prese in rassegna senza pretesa di essere approfondite.

Posticipazione addRelated

Per poter protocollare l'intera unità documentale è necessario che i vari allegati siano relazionati (related) al documento principale. Le relazioni possono essere impostate contestualmente alla creazione del documento (createDocument, poi addRelated), come proposto in questo documento, oppure collegati subito prima della chiamata di protocollaDocumento. In quest'ultimo caso si può dare come assunto che l'unità documentale sia composta da tutti i documenti contenuti nella cartella.

DOC/ER anche per versamento delle pratiche

Richiederebbe l'implementazione anche delle operazioni sui fascicoli, ma permetterebbe di pianificare l'abbandono di Alfresco.

---

[17] In GDA gruppi e utenti non vengono mai cancellati.

[18] Si è ad esempio ipotizzata un'autenticazione con password identica per tutti, o ricavabile sulla base dello username (es. username123); questo approccio è da considerarsi temporaneo, e non considererebbe l'accesso diretto dell'utente a DOC/ER.

FORNITURA

Introduzione

In mancanza di un capitolato tecnico adeguatamente dettagliato, sarà necessario che il fornitore presenti allegato al preventivo una distinta della fornitura per specifiche. In questo capitolo verrà proposta un'ipotesi in tal senso.

Questo l'elenco dei componenti forniti:

Componente	Descrizione
Explorer di documenti	Applicazione web per la gestione dei documenti di protocollo
Libreria wrapper DOC/ER	Libreria per implementare nel client GDA alcuni metodi DOC/ER
Utilità di sincronizzazione	Utilità per sincronizzare le anagrafiche GDA con DOC/ER

Explorer di documenti

Si tratta di un'applicazione web studiata per poter operare sui documenti in DOC/ER collegati ad un protocollo.

La pagina è accessibile attraverso un url contenete un folderId identificativo della cartella predisposta per il protocollo (al primo accesso dovrà essere creata, vedi più sotto).

La UI si presenta con l'elenco dei documenti versati, e una toolbar per la gestione delle seguenti funzioni:

- versa un nuovo documento (viene richiesto il file e un abstract);
- esegui il download del documento selezionato;
- versa una nuova versione del file selezionato (viene richiesto il file della nuova versione);
- cancella il documento selezionato (viene richiesta una conferma);
- protocolla e archivia il documento selezionato (viene richiesta una conferma)[19].

Libreria client DOC/ER

Si tratta di una libreria pensata per poter implementare alcune funzionalità di DOC/ER direttamente da un applicativo client. Può essere utilizzato nel client GDA in sostituzione del modulo AlfrescoHelper.

A differenza di AlfrescoHelper, l'istanza non viene creata nel contesto di una cartella, ma l'identificativo dell'eventuale cartella o documento su cui operare viene passato direttamente ai metodi.

I metodi implementati sono i seguenti:

[19] Il documento selezionato diventa principale, gli altri allegati e related.

- createFolder(): crea una nuova cartella e restituisce il folderId corrispondente[20];
- children(String folderId): restituisce una lista di mappe con le coppie chiave/valore di tutti i metadati disponibili nel profilo documento dei documenti contenuti nella cartella[21];
- numberOfDocument(String folderId): restituisce il numero di documenti contenuti in una cartella;
- getAllVersions(String docId): restituisce una lista con le versioni del documento[22];
- createDocument(String folderId, String name, byte[] content, String title, String description): crea un nuovo documento nella cartella;
- createVersion(String docId, byte[] content): crea una nuova versione per il documento;
- deleteDocument(String docId): cancella il document;
- getDocument(String docId): restituisce il file del documento come binario;

Note:

I metodi esposti da questa libreria sono implementati internamente tramite le chiamate ai ws esposti da DOC/ER, da cui ereditano conseguentemente caratteristiche e regole. Solo a titolo di esempio, l'efficacia dei metodi createVersion o deleteDocument dipendono dallo stato archivistico impostato da DOC/ER.

Non esiste un metodo copyDocument. Nei contesti in cui è richiesto, sarà necessario eseguire getDocument e createDocument.

Non esiste un getDocumentStream (che restituisce un InputStream). Dove richiesto dovrà essere creato a partire dal byte content.

Guida interventi in GDA

La libreria client DOC/ER dovrà essere usata in rimpiazzo dell'AlfrescoHelper in alcuni contesti del client GDA. Nella tabella successiva vengono elencati i punti individuati, e indicato per ciascuno di essi quali metodi della libreria potranno essere usati:

Metodo client GDA	Note
FormPubblicazioneRiva.pubblicaOra	children, getDocument: non esiste un titolo del file; il documento principale va individuato da TIPO_COMPONENTE; lo stream va generato a partire dal binario.

[20] Il client GDA dovrà registrare questo folderId per poter successivamente accedere alla cartella con l'explorer o per invocare su essa altri metodi della libreria.

[21] Internamente verranno usati i metodi getFolderDocuments e getProfileDocument

[22] In DOC/ER le versioni di documento non hanno un profilo diverso dal documento stesso.

Metodo client GDA	Note
FormDetermina.createDocument	createDocument
RuleSet.eval	children per recuperare il DOCNAME
PraticaUtil.protocollaPratica	createFolder
SimpleWorkflow.createBindings	children per recuperare il DOCNAME, libreria da passare al contesto dello script
FormProtocollo.inviaPec	children, getDocument: lo stream va generato a partire dal binario
FormScrivania.refreshInfo	numberOfDocuments
PubblicazioneUtil.pubblicaProtocollo	children, getDocument: i documenti da generare nella folder Alfresco della pubblicazione dovranno essere generati con AlfrescoHelper

Note:

Gli script groovy eseguiti nei passi di workflow hanno un riferimento ad AlfrescoHelper tra i bindings. Il codice può essere modificato per includere un riferimento anche alla libreria client DOC/ER, ma gli script andranno revisionati.

Il webservice di protocollazione attuale

La versione attuale di GDA ha un servizio di protocollazione che viene utilizzato da alcune applicazioni. Si tratta di un servizio rest implementato con jax-rs, che consiste in una prima chiamata che produce la registrazione di protocollo, e con una successiva chiamata che permette di allegare un file. Quest'ultimo dovrà essere adattato per andare in scrittura sul documentale tramite l'interfaccia DOC/ER.

Utilità di sincronizzazione

Verranno date indicazioni su come utilizzare la libreria client DOC/ER in script per la sincronizzazione delle relazioni tra utenti e uffici.

COSTI[23] E TEMPI

Analisi, sviluppo, produzione

La realizzazione del progetto è suddivisa in tre fasi distinte. La prima fase di analisi e progettazione porterà alla stesura e consegna di un documento di specifiche operative, per validazione da cliente. La successiva fase di sviluppo si concluderà con la messa in test della soluzione, mentre la fase finale con la messa in produzione e validazione.
Il progetto si intende completo e validato dopo due settimane dalla messa in produzione, a meno che non insorgano anticipatamente problemi tali da doverlo rimuovere dalla produzione.

Descrizione	Costo
Analisi e progettazione	€ 3.200,00
Sviluppo e messa in test (stima su presente analisi)	€ 4.800,00
Messa in produzione e validazione (stima su presente analisi)	€ 2.400,00
Totale	**10.400,00 €**

Al termine di ogni fase verrà emessa relativa fattura con scadenza a 30gg.

Tempi

I tempi consegna[24] sono di 120 giorni a partire dalla messa a disposizione dell'installazione DOC/ER, della documentazione necessaria, e del supporto operativo da parte di personale qualificato sulla piattaforma DOC/ER. In caso di situazioni tali da portare a una revisione della stima, questa verrà riportata fase per fase.

Note

Costi e tempi sono stimati sulla base del presente documento d'analisi.

Il committente ha espresso la volontà di affidare in un'unico incarico tutte le fasi di analisi, sviluppo e produzione, senza affidare un incarico di analisi e stesura di un capitolato su cui basare la richiesta di offerta.

L'offerta proposta sarà quindi vincolata all'accettazione della presente analisi, che andrà a costituire il capitolato di fornitura sul quale è formulato il preventivo, e primo passo concluso.

[23] I costi per le varie fasi rappresentano solo una prima stima.

[24] Qui si intende per "consegna" la messa in produzione di una versione del software adeguata per la messa in produzione, completa quindi nelle sue componenti principali. Eventuali mancanze e/o difetti che non dovessero essere considerati tali da escludere la messa in produzione, e potranno essere sistemati o completati prima della validazione.