LINEE GUIDA

PROGETTAZIONE E SVILUPPO

Indice

2 CONFIGURAZIONE AMBIENTE DI SVILUPPO LOCALE

L'ambiente per lo sviluppo del software prevede i seguenti strumenti/elementi in ambiente MS Windows 10:

1. ClesiusDeveloperSystem in I:\shared_resourses (configurazioni utente);
2. Eclipse (con eventuali plug-in);
3. Squirrel;
4. Open Office;
5. TortoiseSVN;
6. Jasper Studio per l'editing dei report JasperReports.
7. Gimp
8. Tar Gzip
9. MS Sql server 2005 client;
10. MS Office Professional;
11. Driver odbc per la connessione ai database;
12. WinSCP3 / Putty per accede agli ambienti di preproduzione/produzione Linux;
13. OpenSSH per la connessione remota criptata;
14. CygWin;
15. MS Terminal server client per accedere ad ambienti di preproduz./produzione Windows;
16. Notepad++ per editing HTML;
17. PSPad o Scintilla per l'editing di file ASCII;
18. IE
19. Firefox
20. Chrome

3 CLESIUS DEVELOPER SYSTEM

Le funzioni del tool ClesiusDeveloperSystem (I:\shared_resourses) sono:

1. l'installazione di Tomcat sul proprio disco locale di sviluppo. Il tool ClesiusDeveloperSystem permette di scaricare la versione di pre-produzione del codice di un ambiente (ISEE, ICEF), si occupa di fare l'unione di Tomcat e Apache e di configurare il Tomcat locale. Un apposito pulsante permette, dopo aver scelto il database a cui collegare il Tomcat locale, di avviare il Tomcat stesso, di fermarlo, di pulire i log e di configurare il web.xml. Per il corretto funzionamento del Tomcat 7 è necessaria la presenza del JDK 1.7 sul computer locale;
2. La connessione con il protocollo SFTP ai server di pre-produzione e a quelli di produzione per le figure aziendali con le necessarie autorizzazioni;
3. L'allineamento della produzione con la pre-produzione e della pre-produzione con la produzione di tutti gli ambienti, con suddivisione tra ambiente Tomcat, Apache codice e Apache documentazione.
4. La creazione del jar evol5 firmato a partire dal codice commitato sul SVN.

I database a cui è possibile connettersi con il tool ClesiusDeveloperSystem sono:
PRODUZIONE

Nome	Connessione	IP	DB connessi
ICEF	VPN	172.31.16.65	ClesiusICEF

SVILUPPO E PREPRODUZIONE

Nome	IP	DB connessi
ICEF_PREPROD	192.168.30.244:1433	ClesiusICEF (copia produzione)
ICEF_DEV	192.168.30.244:1433	ClesiusICEF_DEV (sviluppo)

4 ECLIPSE

Il software di riferimento per lo sviluppo del codice java, jsp, xml è Eclipse. Per installare Eclipse è sufficiente copiare i file dell'ultima versione presente in I:\nobackup\sw\Internet\dev\Java SW Development Environment\Eclipse oppure è possibile scaricarlo da internet dal sito http://www.eclipse.org/ e scompattarlo in una directory locale. Una volta installata la versione che si ritiene adatta allo sviluppo è possibile installare i plug-in più utilizzati e stabili da http://www.eclipse.org.

5 CLIENT PER I DATABASE

E' possibile utilizzare SQuirreL per connettersi a database di tipo MS SQLSever, Oracle e PostgreSQL. Oltre a questo tool è possibile utilizzare dei client che girano sotto Windows a cui si fa riferimento nei paragrafi successivi e che sono specifici per tipologia di database.

5.1 MS SQLServer

La versione da installare è Microsoft SQLServer 2005 client. L'installazione è possibile tramite i cd originali. Sfruttando la retrocompatibilità, nonostante il server sia fermo alla versione 2005, è possibile scalare con il client sino alla versione 2014.

5.2 MS Access

La versione da installare è Microsoft Access 2003. L'installazione è possibile tramite i cd originali di Office 2003. MS Access viene utilizzato sia per la connessione ai database dei dati sia per lo sviluppo delle maschere attraverso una connessione di tipo ODBC (sono necessari i relativi driver ODCB).

5.3 OpenOffice Base

OpenOffice Base viene utilizzato per la connessione ai database dei dati attraverso una connessione di tipo JDBC

5.4 Squirrel

Utilizzare SQuirreL come client SQL per connettersi a database di tipo MS SQLSever, Oracle e PostgreSQL.

6 OPEN OFFICE E OFFICE

I programmi di editing di testo e foglio di calcolo possono essere installati seguendo le seguenti modalità. In ambiente Linux la suite di riferimento è OpenOffice scaricabile dall'indirizzo http://it.openoffice.org/ mentre in ambiente Windows è possibile utilizzare MS Office comprensivo di Access.

7 SERVIZIO SVN

7.1 CLIENT

7.1.1 Client desktop

I client SVN da utilizzare sono:

- TortoiseSVN per interagire con il server SVN direttamente dall'albero delle cartelle locali: di fatto è un plugin per l'esplora risorse di windows che ne estende le funzionalità permettendogli di dialogare con un SVN server remoto;
- Eclipse: per interagire con il server SVN direttamente dall'interno dell'ambiente di sviluppo.

Ogni programmatore avrà un account sul server SVN per poter gestire la documentazione di propria competenza relativa ai progetti. Il Server SVN ha IP 192.168.30.208.

7.1.2 Client web

Il contenuto del server SVN è consultabile direttamente via browser senza necessità di client specifici installati.
L'accesso avviene attraverso l'URL http://192.168.30.208/clesius ovviamente a seguito di autenticazione.
E' possibile sfruttare un accesso in sola consultazione e ricerca, senza la possibilità di scaricamento in locale, attraverso l'URL http://192.168.30.208/webSVN che non necessita di autenticazione.

7.2 SERVER

Il server SVN è una macchina virtuale linux con IP 192.168.30.208.

8 CONNESSIONI SSH CON SERVER REMOTI

Tutti i server di produzione e preproduzione sono basati su OS Linux e quindi è necessario avere dei client che supportano il protocollo ssh per gestirli ed usarli da remoto.

8.1 Putty e winscp

Putty è un client ssh (vedi www.putty.org) e WinSCP è un client SCP (vedi https://winscp.net)

8.2 Cigwin

8.2.1 Ssh

Client ssh (vedi https://www.cygwin.com)

8.2.2 sftp

Client SCP (vedi https://www.cygwin.com)

9 RDESKTOP E MS TERMINAL SERVER

Il terminal server client permette di gestire gli ambienti di produzione / preproduzione MS Windows (ICEF) consentendo di accede in emulazione terminale al server. I server a cui è possibile connettersi sono i seguenti:

- 192.168.30.244: server database Icef Preproduzione;
- 172.31.16.65: server database Icef Produzione;
- 192.168.30.1: client di test per desktop con OS in versione XP.

Il client che permette invece di gestire i server Linux in emulazione terminale è RDesktop.

10 CREAZIONE E GESTIONE DI STAMPE E REPORT

10.1 Ireport

Ireport è il programma utilizzato per lo sviluppo dei jasper. La versione compatibile con le librerie in produzione è la 3.0.0. Tali librerie sono mantenute in produzione per la gestione di quanto sviluppato precedentemente al rilascio di Jaspersoft Studio.

10.2 Jaspersoft Studio

Jaspersoft Studio è un report designer per JasperReports e JasperReports Server open source e basato su Eclipe

11 EDITOR FILE HTML E FILE ASCII

Per gli ambienti Windows (ed eventualmente Linux utilizzando Wine come emulatore) l'editor per i file html è Notepad++, mentre per gli ambienti Linux mc.
Per i file ASCII si utilizza PSPad oppure Scintilla.

12 EDITOR DI IMMAGINI

Il programma editor di immagini utilizzabile è Gimp (ultima versione disponibile), la cui versione si trova sia per SO Windows che per SO Linux. Uno degli usi più comuni di GIMP è per preparare immagini che devono apparire meglio possibile su un sito Web . A tal fine si consulti https://docs.gimp.org/it/gimp-using-web.html

13 PROGRAMMI DI COMPRESSIONE E DECOMPRESSIONE

I programmi per comprimere / decomprimere i file che possono essere utilizzati sono, per l'ambiente Linux tar e gzip (si può utilizzare il comando slipt per suddividere i file e rendere più facile l'eventuale upload / download degli stessi), per l'ambiente windows 7-zip.

14 BROWSER

I browser da utilizzare per la navigazione e da utilizzare per i test dell'applicazione sono:

- Internet Explorer 8 (SO Windows XP) o Internet Explorer ultima versione (OS Windows 10);
- Firefox ultima versione (SO Windows + SO Linux);
- Chrome ultima versione (SO Windows + SO Linux).

15 CICLO DI PRODUZIONE DEL SOFTWARE

15.1 In locale

Il ciclo di sviluppo del software standard prevede i seguenti passi:

- Sincronizzare il codice presente sul proprio disco di sviluppo locale con il codice presente sul relativo server di pre-produzione attraverso il ClesiusDeveloperSystem.
- Importare o sincronizzare i propri sorgenti con quelli presenti su SVN (se necessario). Per fare ciò si utilizza il client SVN installato, comando Update;
- Creazione e/o modifica dei file sorgenti con Eclipse o/e l'editore html installato;
- Per i file che necessitano di essere compilati, compilazione e installazione degli stessi nel proprio ambiente di test locale. Lo sviluppatore può prevedere una compilazione automatica nella directory corretta o l'utilizzo di script ANT per eseguire l'installazione. Si ricorda a tal proposito che sino a quando non verrà abbandonata definitivamente la MS Java virtual machine è necessario che le classi java lato client siano compilate con un jdk 1.8;
- Modifica alla struttura e alla configurazione del database di sviluppo *_DEV;
- Connessione con il ClesiusDeveloperSystem al database _DEV dell'installazione per la quale di sta sviluppando;
- Test del codice ed eventuale debug.

Una discussione a parte vale per la modifica relativa alle classi di trasformazione, jar e net di calcolo. In questi casi, dopo un test con il database di sviluppo è necessario eseguire il seguente Test connettendo il codice dell'ambiente di sviluppo locale con il database di produzione:

- Utilizzo del tool Test Console per la verifica delle differenze tra i documenti di test preinstallati;
- Verifica ed eventuale validazione dei test con creazione del test di verifica.

15.2 Delivery in pre-produzione ad eccezione delle librerie

L'ambiente di pre-produzione è comune a tutti gli sviluppatori e serve per effettuare i test finali in un ambiente identico a quello di produzione. Ciascun utente prenota una sessione di lavoro che gli permetterà di avere accesso in maniera esclusiva al server per testare le modifiche che intende installare in produzione. Dal momento che chi attiva una sessione di lavoro blocca ai colleghi l'accesso al server di pre-produzione, è importante che la sessione venga sempre chiusa al termine dei test e che questi siano predisposti in modo da occupare il server per il minor tempo possibile.

Per rilasciare la versione del codice in pre-produzione:

- Allineamento del codice della pre-produzione con il codice in produzione;
- Allineamento del database *_preprod della relativa installazione;
- Copia manuale dei file compilati in locale nell'ambiente di pre-produzione relativo mediante WinSCP;
- Test della configurazione;
- Aggiornamento dei sorgenti sul SVN (java, jsp, xml) utilizzando il programma TortoiseSVN (o programma analogo) comando commit;
- Predisposizione della documentazione tecnica;

Per il rilascio di nuove versioni delle librerie jar Clesius si consulti il paragrafo Aggiornamento delle librerie Clesius.

15.3 Delivery in produzione

Per il rilascio delle modifiche in produzione:

- Allineamento del database di produzione della relativa installazione;
- Allineamento della produzione con la preproduzione. Si ricorda che gli aggiornamenti lato Tomcat avranno effetto a partire dal successivo riavvio del Tomcat, mentre le modifiche lato Apache avranno effetto a partire dal reset della servlet di gestione dello stream download. Per i motivi appena citati è necessario che il responsabile tecnico valuti correttamente il momento del delivery in produzione;
- Attivazione del test di validazione, in caso di esito positivo avvio del servizio;
- Avviso alla struttura di supporto delle modifiche installate.

16 I SERVER DI PREPRODUZIONE

I server di pre-produzione sono virtualizzati con VMware su di un unico server fisico che risponde alle chiamate in maniera del tutto analoga ai server originali. I server di pre-produzione linux/unix sono raggiungibili con WinSCP3 o OpenSSH con cui si possono effettuare gli aggiornamenti.

- 192.168.30.224 Apache INFOTN;
- 192.168.30.225 Tomcat INFOTN;

17 MODIFICHE DIRETTAMENTE IN PRODUZIONE

Le modifiche che possono essere effettuate direttamente in produzione riguardano la modifica/aggiunta/eliminazione (in alcuni casi) di record del database presenti nelle seguenti tabelle:

R_Enti;

R_Uffici;

R_Utenti;

R_responsabili;

le tabelle di configurazione delle news, faq, documenti, normativa.

In tutti gli altri casi è vietato effettuare gli aggiornamenti direttamente in produzione senza passare dalla preproduzione.

18 INDICAZIONI PER I JAVADOC

La creazione dei javadoc è possibile da eclipse utilizzando la funzione Project -> Generate Javadoc. E' buona norma seguire le direttive standard utilizzate da JAVADOC per creare i propri java. Ricordare che per andare a capo è necessario utilizzare
 al fine di ottenere una corretta impaginazione di Javadoc.
Per ogni classe occorre indicare i seguenti tag:

```
/**
 * Commento esplicativo di cosa fa la classe
 * @version versione
 * @author Nome Cognome
 */
```

In caso di modifiche inserire il nome e cognome dell'autore delle modifiche e relativa versione. Esempio:

```
/**
 * Commento esplicativo di cosa fa la classe
 * @version versione1 - cosa è cambiato con la versione
 * @author Nome Cognome , AltroNome AltroCognome (versione1)
 */
```

Per ogni dichiarazione di variabile occorre indicare cosa rappresenti. Ad esempio:

```
/** Nr.di componenti */
int ncomp=0;
```

Per ogni metodo occorre indicare i seguenti tag:

```
/**
 * Cosa fa il metodo in maniera abbastanza puntuale
 * @param nomeparametro1 - cosa rappresenta il parametro1
 * @param nomeparametro2 - cosa rappresenta il parametro2
 * ...
 * @return l'eventuale valore di ritorno
 * @throws ServletException,IOException,... in base a quali siano state dichiarate
 * @see percorso completo del metodo/classe che bisogna consultare. Esempio:
it.clesius.activator.OptionPanel#getTitoloTab() collega il metodo getTitoloTab() della classe
OptionPanel
 */
```

Il tipo di variabile NON VA INDICATO IN @param poichè va indicato solo il nome.
NON AGGIUNGERE segni di PUNTEGGIATURA a tag e a nomi di variabili.
Tra metodi, classi e variabili va lasciata SOLO UNA RIGA VUOTA.

Evitare l'uso eccessivo di righe vuote all'interno di un metodo. E' preferibile inserire un commento invece di righe vuote.
Il commento deve precedere il metodo, la classe o la variabile cui si riferisce senza alcuna riga di spazio.
Non utilizzare tag non permessi (esempio: @TODO in un commento oppure @author in un metodo).

19 INDICAZIONI PER FILE JAVA

La prima riga di un file java deve avere il riferimento al package. Successivamente vanno indicati gli import utilizzati (evitare di inserire import che non vengono utilizzati o utilizzati parzialmente). Quindi vanno le classi con i relativi metodi e variabili.
Specificare per ogni classe/metodo/variabile la sua visibilità: public, protected, private.
Ricapitolando la struttura di un file java sarà tipo:

```
package nomepackage;
import importpackage;
...
import importclass;
/**
 * Commento sulla classe
 * @version ...
 * @author ...
 */
public class nomeclasse extends ... implements ...{

        /** commento della variabile */
        public String var="";

        /**
         * Commento sul metodo
         * @param uno commento parametro1
         * @param due commento parametro2
         * @return commento valore di ritorno
         * @throws Exception in caso di errore
         * @see ...
         */
        private String nomeMetodo(int uno, boolean due) throws Exception {
        }
}
```

Normalmente gli IDE (Eclipse) aiutano a costruire il tutto in maniera conforme a quanto qui esposto.

20 CONVENZIONE NOMI E PERCORSI

20.1 Tabelle, viste e stored procedure del database

Per quanto riguarda le tabelle presenti nel database dei dati vanno osservate alcune regole che riguardano la nomenclatura delle stesse:
Il nome delle tabelle tp_ e R_, sia generiche che specifiche per una politica, deve essere al plurale mentre il nome dei campi della tabella deve essere al singolare. Esempio: la tabella che raccoglie la nomenclatura degli investimenti si chiamerà tp_investimenti, le colonne della tabella saranno invece al singolare ID_tp_investimento (l'identificativo) e tp_investimento (la descrizione);
Il nome delle tabelle specifiche di una politica in cui o verranno salvati i dati, o ci saranno delle tipologie (tabelle tp_) o delle configurazioni (tabelle R_), va preceduto dal suffisso della politica stessa e da "_". Esempio: la tabella dei dati dell'assegno di studio per le scuole nell'ICEF dovrà essere del tipo SCUOLE_Dati, la tabella che raccoglie la nomenclatura delle classi di frequenza dovrà essere del tipo SCUOLE_tp_classi e la tabella che raccoglie il nome delle scuole dovrà essere del tipo SCUOLE_R_Scuole;
Il nome delle viste sul database dovrà essere del tipo v_[nome della politica]_[anno della politica]_[nome vista]_[iniziali di chi ha creato la vista], dove [nome della politica] deve rispecchiare il nome della politica così come indicato anche nelle tabelle del DB, [nome vista] è la descrizione in breve di cosa fa la vista, e [iniziali di chi ha creato la vista] sono le iniziali dell'autore della vista;
Il nome delle StoredProcedure dovrà seguire lo stesso schema indicato sopra per le viste solo preceduto da "sp" invece che da "v", quindi sarà del tipo sp_[nome della politica]_[anno della politica]_[nome vista]_[iniziali di chi ha creato la vista].

20.2 StoredSQL dei mask, ActionButton e DBDirect

Gli StoredSQL dei mask devono seguire alcune regole che riguardano la loro nomenclatura. Il nome della classe deve essere composto in questo modo StoredSQL[tipo documento][nome politica][zona di attuazione*][anno*] dove:
[tipo documento] riguarda la natura del documento (domanda, dichiarazione, scheda);
[nome politica] è la descrizione della politica a cui si riferisce (domande) o il tipo di documento negli altri casi (es: per le dichiarazioni è il tipo di dichiarazione);
[zona di attuazione*] è l'ambito territoriale di attuazione, ad esempio Rimini, Veneto, Roma e così via (opzionale);
[anno*] è l'anno di attuazione della politica (opzionale).
Per quanto riguarda i DBDirect valgono le stesse regole esposte sopra per gli StoredSQL quindi i nomi delle classi dei mask e del DBDirect relativo differiranno solo per il prefisso StoredSQL o DBDirect.
Gli ActionButton si dividono in specifici o generici. Gli ActionButton specifici sono quelli che si riferiscono ad una sola maschera e devono essere posizionati nello stesso package della maschera stessa. Il nome dell'ActionButton deve essere così composto [nome politica][zona di

attuazione*][anno*][descrizione action button] dove i primi tre elementi del nome devono essere uguali a quelli del mask a cui si riferisce il bottone (classe).

20.3 Mdb dei mask

I file mdb di sviluppo dei mask sono posti in I:\evolution_mask_mdb\, all'interno di questa directory la struttura è la stessa che si segue per quanto riguarda il posizionamento nei package delle classi relative ai mask.
Per facilitare la relazione tra l'mdb di sviluppo del mask e la classe StoredSQL relativa, i nomi degli mdb devono essere composti in questo modo ClesiusMask[tipo documento][nome politica][zona di attuazione*][anno*] in analogia a quanto indicato sopra per gli StoredSQL.
Nella stessa directory in cui si trova il file mdb di un mask è obbligatorio salvare la documentazione in formato html che si genera automaticamente al momento della creazione del codice java relativo al mask considerato.

21 DOVE TROVARE LE LIBRERIE DA INCLUDERE NEI PROGETTI ECLIPSE

Per quanto riguarda la gestione delle librerie necessarie per i vari progetti Eclipse è stato deciso di utilizzare un repository comune situato sul primary domain controller e accessibile come servizio intranet che utilizza la struttura del progetto Maven (http://maven.apache.org/). Nella creazione di un progetto con Eclipse è quindi possibile linkare le librerie necessarie al progetto stesso direttamente puntando alle librerie presenti in rete.

21.1 Librerie di terze parti

La struttura del repository è creata in modo tale che la libreria Y della società X nella versione Z.z sarà disponibile nella directory T:\jtx\maven\repository\X\Y\Z.z. Quindi per le librerie di terze parti sarà possibile trovare la versione che interessa nella relativa directory.

21.2 Librerie Clesius

Il discorso cambia per quanto riguarda le librerie Clesius (es: evol5.jar). Al momento del rilascio di una nuova versione delle librerie Clesius nella directory del Tomcat su T dove sono installate si crea sia il file con il nome standard che il file seguito da delle cifre che ne identificano la data di compilazione.

21.3 Ulteriori potenzialità di Maven

La struttura sopra descritta si basa su un progetto, chiamato Maven, che non si occupa solamente di essere un repository delle varie versioni delle librerie, sia di terze parti che proprietarie Clesius, ma si occupa anche di gestire le dipendenze tra le varie librerie. In un file .pom presente nella directory della versione Z.z della libreria Y vengono descritte tutte le dipendenze che la libreria Y versione Z.z ha con altre librerie. Utilizzando un plugin di Eclipse è possibile utilizzare il motore di Maven e fare in modo che vengano importate nel progetto tutte le

librerie con dipendenze con quelle che stiamo utilizzando e se tali librerie non sono presenti nel repository si occupa di scaricarle da internet nella directory corretta.

22 DOCUMENTAZIONE TECNICA DEI PROGETTI

La documentazione tecnica di un progetto deve essere posta nel gestionale ISO 9001 –> documentazione progetti. La suddivisione dei progetti è la seguente:

- macroarea
- committente
- commessa
- servizio
- produzione
- doc_tecnica