Finsoft srl

Osai AS SpA

Luglio 2020 – Febbraio 2021

# Piattaforma reportistica ore progetti RD

## Abstract

OSAI ha la necessità di dichiarare a terzi le ore che i suoi dipendenti spendono sui progetti RD (ricerca e sviluppo).

È necessario uno strumento che generi una base dati a partire dalle esportazioni Excel dei LUL (tipo fogli ore) e dai rapportini dei dipendenti, e generi reportistica elaborando i dati importati con dati presenti su un’altra base dati SQL Server esterna (Panthera).

## Architettura

Ipotizziamo di sviluppare un’applicazione web su stack Linux+Apache+MySQL+PHP.

## Funzionalità previste

* Login tramite LDAP
* Importazione dati dai fogli Excel LUL
* Immissione anagrafica di progetto
* Esportazione rapportini dipendente Excel da compilare
* Importazione rapportini dipendente Excel compilati
* Ristampa rapportini dipendente Excel compilati
* Generazione due report budget
* Generazione dati di test

## Descrizione del processo

Il processo viene eseguito con cadenza mensile.

1. Un addetto esporta, a inizio mese, i rapportini per tutti i dipendenti, precompilati con date, festività, e WP (work package) associati al dipendente.
2. Ogni dipendente dovrà compilare il rapportino indicando, per ogni WP e per ogni data, il numero di ore che egli ha lavorato quel giorno su quel WP.
3. Ogni dipendente dovrà consegnare a fine mese il rapportino in forma elettronica all’addetto, e dovrà anche consegnarne copia cartacea firmata.
4. L’addetto caricherà a sistema tutti i rapportini mediante l’apposita funzione di importazione.
5. Quando arrivano i dati del LUL (presumibilmente diversi giorni dopo fine del mese), l’addetto deve importare a sistema tutti i dati LUL per tutti i dipendenti.
6. A questo punto, sarà possibile generare i due report del budget.

In caso di controlli sarà possibile ristampare i rapportini. In questo caso i rapportini usciranno dal sistema riportando le ore dichiarate dal dipendente.

## Anagrafiche di Panthera

Alcuni dati necessari ai calcoli (in particolare i costi) sono presenti sul database SQL Server del software gestionale Panthera.

I dati di Panthera **non** saranno duplicati nel database locale, salvo dove indispensabile. Vengono acceduti solamente quando necessario.

I dati che possono interessarci sono: Anagrafica dipendente (nome, cognome, matricola) e costo orario dipendente di tipo “ricerca&sviluppo”.

Tutti i dipendenti sono censiti in Panthera come risorse di livello matricola, e i costi orari sono salvati all’interno della risorsa stessa. I costi possono variare nel tempo, e sono ivi storicizzati. Esistono attualmente tre diversi “tipi di costo”.

Tutte le tabelle di Panthera verranno filtrate per **ID\_AZIENDA=’001’**.

## Funzioni disponibili

### Griglia dei progetti.

Occorre una pagina che mostri l’elenco dei progetti presenti a sistema. Dalla griglia dei progetti essere possibile creare, modificare, eliminare i progetti. Deve essere prevista una funzione di ricerca per acronimo, titolo e descrizione.

### Maschera di immissione/modifica del progetto

I dati da imputare sono i seguenti:

* Progetto
  + Acronimo
  + Titolo
  + Grant Number
  + Abstract (facoltativo)
  + elenco dei workpackage
    - numero
    - titolo
    - descrizione (facoltativa)
    - data inizio
    - data fine
    - monte ore
    - elenco persone assegnate
  + monte ore totale (vincola monte ore dei WP)
  + data inizio (vincola data inizio dei WP)
  + data fine (vincola data fine dei WP)
  + costo medio uomo mensile (a fini budget)
  + “tipo costo” di Panthera da utilizzare nei consuntivi (menù a tendina)
  + Supervisor (project manager)
  + elenco altre spese varie
    - descrizione
    - importo
    - tipologia (campo tabellato)

La maschera dovrà recuperare dal database Panthera nomi e matricole dei dipendenti, che poi l’utente potrà associare liberamente ai vari WP. Dovrà recuperare anche l’elenco dei “tipi costo” disponibili secondo la relativa validità temporale.

Deve essere possibile creare, modificare, eliminare tutte le voci delle sotto-tabelle (WP, dipendenti, voci di spesa).

Sulla maschera potremmo fare apparire, in sola lettura, per comodità dell’utente, alcuni campi statistici calcolati sul momento, quali:

* il “budget” del progetto, calcolato come prodotto del “costo medio uomo” per il “monte ore totale”
* il “totale delle spese”

### Importazione LUL

Occorre una pagina da cui uploadare il/i file contenenti i dati di presenza dei dipendenti.

Il formato di esportazione non è ancora noto, presumibilmente XLSX.[[1]](#footnote-1)

I dati contenuti saranno salvati in un’opportuna tabella del database locale. Tale tabella dovrà salvare queste informazioni: matricola dipendente, data, numero ore ordinarie lavorate. (Lo straordinario è riportato separatamente e verrà ignorato).

Occorre prevedere che i dati possano essere ri-caricati in caso di errore. In questo caso, dopo opportuna conferma dell’utente, i dati preesistenti saranno cancellati e sostituiti.

Non sappiamo ancora quali dati siano qui presenti, ipotizziamo che per ogni anagrafica del dipendente siano riportate le ore lavorate ogni giorno. Non ci interessano gli orari.

### Esportazione rapportini dipendente Excel da compilare

Questa funzione prende come unico parametro il mese di cui si intende eseguire la generazione dei rapportini. Genera un file ZIP contenente tanti file Excel, uno per ogni dipendente.

Ogni foglio dipendente riporterà in alto le date del mese, a sinistra l’elenco di tutti i progetti/WP associati all’utente.

Sulla seconda riga saranno indicate le festività, estrapolate dai LUL (potranno apparire solo in fase di ristampa)

Sulla terza riga saranno indicate le ore effettivamente lavorate, estrapolate dai LUL (potranno apparire solo in fase di ristampa)

Sulla quarta riga saranno indicate le ore rendicontate per tale giorno (tot ore)

Sulla quinta riga saranno indicate le ore rimanenti (ore lavorate – ore rendicontate)

Esempio:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | … |
|  |  |  | festivo | assente |  |  | festivo | … |
| PROJ1 | WP1 |  |  |  |  |  |  |  |
| PROJ1 | WP2 |  |  |  |  |  |  |  |
| PROJ2 | WP3 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |  |

### Importazione rapportini dipendente Excel compilati

Occorre una pagina da cui uploadare il/i rapportini dei dipendenti opportunamente compilati.

I dati contenuti saranno salvati in un’opportuna tabella del database locale. Tale tabella dovrà salvare queste informazioni: codice progetto, codice WP, matricola dipendente, data, numero ore lavorare.

Durante l’importazione, la matricola del dipendente e la mensilità saranno desunti dall’interno del file Excel.

Occorre prevedere che i dati possano essere ri-caricati in caso di errore. In questo caso, dopo opportuna conferma dell’utente, i dati preesistenti saranno cancellati e sostituiti.

### Raccolta data conferma

Al termine del mese sono necessarie le firme di conferma dei dati raccolti: la firma di ciascun dipendente, e la firma del supervisore del progetto.

La piattaforma dovrà quindi raccogliere le due data di firma per ciascun dipendente che abbia lavorato sul progetto nel mese in esame attraverso un form apposito.

Le date di firma devono corrispondere con la presenza della persona in azienda. Nel caso in cui non sia possibile ottenere la firma del supervisore nel mese in corso potrà essere raccolta nel mese successivo. La data di firma del supervisore deve essere successiva alla firma del dipendente.

Sul form verranno suggerite le prime date di firma utili (come aiuto per la validazione) in base a tre criteri:

* ultima presenza nel mese del dipendente
* ultima presenza nel mese del supervisore
* data firma supervisore > data firma dipendente

Nel caso in cui l’ultima presenza nel mese del dipendente sia superiore all’ultima presenza nel mese del supervisore, per quest’ultimo si prenderà la prima presenza utile nel mese successivo.

### Generazione due report budget

Per ciascun progetto, è necessario poter esportare due diversi report, entrambi PDF.[[2]](#footnote-3)

Il primo report, ad uso gestionale, dovrà mostrare la discrepanza tra tempi e costi preventivati (budget) rispetto a tempi e costi consuntivati.

Il secondo, ad uso ufficiale, dovrà mostrare le stesse informazioni con un livello di dettaglio maggiore.

Entrambi i report prendono come parametro il progetto, e come parametro facoltativo la mensilità di cui si vuole analizzare. Se non viene indicata la mensilità si assumerà l’intera durata del progetto.

Entrambi i report dovranno calcolare i dati del consuntivo di progetto. Dovranno accedere alla tabella dei rapportini dipendente per rilevare le ore lavorate da ciascun dipendente su ciascun WP del progetto; e dovranno accedere a Panthera per reperire i costi risorsa relativi (si usa il “tipo di costo” indicato a livello di progetto).

Il primo report dovrà mostrare il confronto tra

* il monte ore del progetto e il numero di ore consuntivate
* il budget del progetto (inteso come monte ore per costo medio risorse) e il costo consuntivato.

Il secondo report dovrà aggiungere il dettaglio per dipendente/WP delle ore lavorate sul progetto, con evidenza del costo orario.

Nel secondo report appariranno dei warning qualora le ore dichiarate dal dipendente nei suoi rapportini siano superiori a quelle derivanti dai LUL.

Ogni report deve contenere le seguenti informazioni:

* logo OSAI
* acronimo del progetto di riferimento
* grant number del progetto
* working person (dipendente)
* supervisor (dipendente)
* data firma working person
* data firma supervisor

### Ristampa rapportini dipendente Excel compilati

La funzione di esportazione dei rapportini dipendente permetterà anche la ristampa nel caso i dati fossero già stati immessi. In questo caso, verrà generato uno ZIP contenente fogli Excel formalmente identici ai precedenti, ma con i dati già compilati.

Sui rapportini non devono in ogni caso apparire costi.

### Generazione dati di test

Per poter eseguire dei test realistici, con molte risorse, si prevede una funzione per la generazione di dati casuali. Questa funzionalità sarà utile solamente in fase di test del software e potrebbe anche essere situata in un programma separato.

Ipotizziamo che tutti i progetti da testare siano già stati compilati, e che i dati LUL siano già presenti.

La funzione prenderà come unico parametro la mensilità per cui si vogliono generare dati.

I dati dovranno essere generati casualmente, ma dovranno tenere conto dei seguenti vincoli. Chiamato il numero di ore che il dipendente i ha lavorato sul WP j nel giorno k, dovremo imporre che:

ore lavorate giornaliere derivate da LUL

monte ore residuo del WP

se il WP j non è ancora iniziato in data k, oppure è già terminato

se il dipendente i non è coinvolto nel WP j

Eventuali dati preesistenti non dovranno essere cancellati

Una soluzione ideale dovrebbe massimizzare la funzione .

Questo è un problema di programmazione lineare intera. Immaginando che ci siano 50 dipendenti da assegnare a 20 WP su 20 giorni lavorativi, dovremo valorizzare 50\*20\*20=20.000 variabili.

Il problema verrà affrontato in maniera naive con un approccio probabilistico. Si noti però che per avere una soluzione ottimale si dovrebbero utilizzare algoritmi standard più complessi quali il *metodo del simplesso*, gli *algoritmi del punto interno*, il *branch and bound*.

## Appendice Tecnica 1 - Algoritmo naive per la generazione casuale dei rapportini

Inizializzo tutte le variabili oppure con gli eventuali dati preesistenti su database

Inizializzo tutte le variabili con il monte ore residuo del WP j, decurtati degli eventuali dati preesistenti su database

Indico con e le date di inizio e fine del WP k

Per ogni data k

Per ogni dipendente i

Identifico quali WP siano ammissibili:

Il dipendente i deve essere associato al WP,

La data k deve essere compresa tra e

Il monte ore residuo deve essere positivo

Per ogni ora che va da 1 a

Estraggo a sorte tra questi quale sia il WP j su cui ha lavorato il dipendente in questa ora

La probabilità con cui estraggo il WP j non dovrebbe essere uniforme; dovrebbe essere:

proporzionale a

inversamente proporzionale a

Incremento

Aggiorno il monte ore del WP appena estratto: decremento

Se rimuovo il WP j dalla lista dei WP ammissibili

Ora ho determinato tutte le variabili per il dipendente i e la data k

Per ottimizzare spannometricamente la soluzione, si potrebbe ripetere l’algoritmo un po’ di volte e scegliere la soluzione ottenuta con maggiore .

## Appendice Tecnica 2 – Ipotesi di struttura tabellare

Prevediamo la creazione delle seguenti tabelle:

**PROGETTI** (chiave ID\_PROGETTO)

**PROGETTO\_WP** (chiave ID\_PROGETTO+ID\_WP)

**PROGETTO\_WP\_RISORSE** (chiave ID\_PROGETTO+ID\_WP+ID\_DIPENDENTE)

**PROGETTO\_SPESE** (chiave ID\_PROGETTO+ID\_SPESA)

**ORE\_PRESENZA** (chiave ID\_ DIPENDENTE+DATA)

**ORE\_CONSUNTIVATE** (chiave ID\_DIPENDENTE+ DATA+ID\_PROGETTO+ID\_WP)

Sarà necessario anche l’accesso ad alcune tabelle di Panthera:

**THIP.TIPI\_COSTO**

**THIP.RISORSE\_COSTI**

**THIP.DIPENDENTI\_V01**

Si filtrerà per ID\_AZIENDA=’001’



1. Nota tecnica: nel caso, possiamo utilizzare la libreria <https://github.com/box/spout> per la lettura dei file. [↑](#footnote-ref-1)
2. Nota tecnica: esistono decine di librerie per la generazione di PDF, verrà utilizzata probabilmente FPDF [<http://www.fpdf.org/>] se il report è sufficientemente semplice, altrimenti Snappy + wkhtmltopdf [<https://github.com/KnpLabs/snappy>]. [↑](#footnote-ref-3)