

# UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Dipartimento di Scienze Matematiche, Fisiche ed Informatiche Corso  
di Laurea in Informatica

## **Creazione di una Dashboard per Human Resource Management in HRTools e Laravel**

Laureando

**Baljinder Singh**

Relatore Prof.

**Riccardo Martoglia**

---

Anno Accademico 2019/20

## **RINGRAZIAMENTI**

Ringrazio Dott. Riccardo Martoglia, mio relatore, per la continua disponibilità e per l'aiuto nell'arco del mio percorso universitario.

Ringrazio Emme Delta per avermi dato la possibilità di sviluppare il presente progetto durante il mio tirocinio.

E un ringraziamento va soprattutto alla mia famiglia e a tutti gli amici e colleghi universitari che mi hanno sempre appoggiato e aiutato.

# PAROLE CHIAVE

Database

Laravel

Human Resources

PHP

Javascript

## Table of Contents

Laureando.....	1
<b>Baljinder Singh</b> .....	1
Relatore Prof. ....	1
<b>Riccardo Martoglia</b> .....	1
Anno Accademico 2019/20.....	1
<b>RINGRAZIAMENTI</b> .....	2
Parte I .....	8
Analisi iniziale .....	8
Capitolo 1 .....	9
Analisi degli Obiettivi.....	9
1.1 – Introduzione.....	9
1.2 – Collaborazione .....	9
1.3 – Richieste.....	10
1.4 – Scopo del Progetto .....	12
Capitolo 2 .....	13
Tecnologie Utilizzate.....	13
2.1 – Introduzione.....	13
2.2 – Database.....	13
2.3 Applicazione Web.....	14
2.3.1 – Front-End .....	15
2.3.2 – Back-End .....	16
2.4 – Ambienti di Sviluppo .....	17
2.5 – Strumenti di Comunicazione .....	17
2.6 – Metodo di Sviluppo Agile .....	18
Capitolo 3 .....	19
Aalisi di Laravel.....	19
3.1 - Introduzione .....	19
3.2 Pattern MVC.....	20
Models: creating our product model .....	21
Controller: creazione del nostro controller .....	23
Views: creazione delle views(viste) dei progetti .....	25

3.3 - Database.....	26
3.4 Seeding .....	27
Parte II .....	29
Progetto e Sviluppo.....	29
Capitolo 4 .....	30
Progettazione.....	30
4.1 - Introduzione .....	30
4.2 Requisiti funzionali.....	30
4.2.1 – Requisiti di Base.....	31
4.2.2 – Requisiti Aggiuntivi .....	32
4.3 - Schema ER di Behavior Set .....	32
4.3 - Progettazione dei widgets per valutazione.....	36
4.4 - Interfaccia.....	41
4.5 Organizzazione Strutturale .....	43
Capitolo 5 .....	45
HRTTools(Human Resource Tools) .....	45
5.1 - Introduzione .....	45
5.2 - Human Resource Management.....	45
5.3 - I moduli di HRTTools.it .....	46
5.3.1 - Anagrafica (People & Organization).....	46
5.3.2 - Activities and Tasks .....	46
5.3.3 – Skills .....	47
5.3.4 Reward .....	47
5.3.5 - Learning.....	48
5.3.6 - Safety .....	48
<b>Conclusioni</b> .....	50
Bibliografia .....	51

## Elenco Figure

- Figura 1.1 Esempio di Tinies Widgets
- Figura 1.2 Esempio di Widgets Group1
- Figura 1.3 Esempio di Widgets Group2
- Figura 2.1 Diagrama Architetturale di Applicazione web
- Figura 2.2 Diagrama Metodo Agile
- Figura 3.1 Diagrama Model View Controller di Laravel
- Figura 4.1 Esempio di un widget di comportamento
- Figura 4.2 Schema scheletro di Modulo Behaviours
- Figura 4.3 Schema ER di Modulo Behaviours
- Figura 4.4 Schema Scheletro Gestione Utenti
- Figura 4.5 Diagramma ER Gestione Utenti: Attributi e Chiavi
- Figura 4.6 Schema Fisico di base di dati
- Figura 4.7 Esempio schede di Valutazione
- Figura 4.8 Esempio di un Behaviour set
- Figura 4.9 Esempio di Behaviour Values
- Figura 4.10 Esempio di un widget di controlla schede compilate
- Figura 4.11 interfaccia hrtools.it
- Figura 4.12 Esempio di struttura delle cartelle
- Figura 5.1 Modulo *rewards*, Esempio di Dinamica retributiva

# Introduzione

La tesi si pone l'obiettivo di analizzare nel dettaglio il tirocinio da me svolto presso l'azienda Emme Delta Consulting di Modena dal mese di Novembre 2020 a quello di Febbraio 2021, per contribuire alla realizzazione del sito [HRTools.it](http://HRTools.it) un sito per Human Resource Management.

Nella prima parte dell'esperienza lavorativa, lo scopo del tirocinio era di imparare nuovi linguaggi di programmazione come php, javascript, vuejs e a scrivere codice per un sito che poggia sul framework Laravel.

L'obiettivo principale consisteva però nella creazione di un modulo da inserire in una applicazione web per la gestione dei dipendenti delle aziende che hanno intenzione di comprare e usufruire dei servizi offerti da Emme Delta Consulti.

Il presente documento è organizzato in cinque capitoli. Il primo capitolo descrive lo scopo del progetto e le varie richieste del cliente. Nel secondo capitolo verranno descritte le tecnologie utilizzate per la realizzazione di questo progetto e l'ambiente di lavoro. Nel terzo capitolo verrà mostrato il famoso framework laravel e il suo modello mvc (il modello di progettazione comunemente utilizzato per lo sviluppo di interfacce utente che divide la logica del programma correlato in tre elementi interconnessi). Nel quarto capitolo sono state spiegate le fasi di progettazione, è descritta struttura del database e dei moduli sul quale sono stati implementati i widgets. Nel quinto capitolo si tratta la potenzialità di [HRTools](http://HRTools.it) e i moduli che offre. Infine, una conclusione degli obiettivi raggiunti con questa esperienza svolta.

# Parte I

## Analisi iniziale



# Capitolo 1

## Analisi degli Obiettivi

### 1.1 – Introduzione

Questo capitolo introdurrà la collaborazione con Emme Delta e descriverà i tratti principali della richiesta del progetto che mi è stato proposto dall'azienda come argomento di tirocinio.

Nella prima parte dell'esperienza lavorativa, gli scopi del tirocinio erano vari: da imparare nuovi linguaggi di programmazione come php, javascript, vuejs fino a scrivere codice per un sito che poggia sul framework Laravel.

Per imparare Php e il framework Laravel ho utilizzato il sito <https://laracasts.com> che mi era stato consigliato dai responsabili dell'azienda.

Una volta imparato tutto, l'obiettivo successivo era di capire tutte i moduli del sito come erano fatti, com'era la struttura del codice e il metodo di progettazione.

### 1.2 – Collaborazione

La realizzazione di questo progetto nasce dalla collaborazione con Emme Delta, come argomento del tirocinio universitario svolto nel corso dell'anno accademico corrente.

Emme Delta dal 2012 è in grado di offrire un sistema integrato di consulenza per l'HR Management orientato alla struttura e ai processi. Emme Delta agisce oggi sullo sviluppo del Knowledge Management attraverso una gestione efficace delle persone in termini di organizzazione, di processi, di sistemi di valutazione, di misurazione delle performance e di sistemi premianti.

Il primo incontro con l'azienda è avvenuto nel Novembre del 2020. Durante la prima giornata ho avuto il piacere di conoscere Emanuele Sartori e Gianpaolo lo Presti il quale, ha successivamente ricoperto il ruolo di tutor aziendale nel corso del mio tirocinio.

## 1.3 – Richieste

Emme Delta mi ha proposto di occuparmi della progettazione di vari moduli per portare tutti i servizi offerti nel Hrtools3 alla nuova piattaforma Hrtools4 che è basato su nuovo framework con una base di dati più ottimizzata per la gestione del sito web che si occupa delle risorse umane.

Inoltre dovevo rendere possibile le importazioni di dati di alcuni modelli attraverso un file di excel (CSV), e le esportazioni di dati in un file di excel per il backup di ogni singolo modulo.

Una volta fatto questo dovevo creare dei widgets per analizzare i dati. I widgets dovevano essere divisi in tre categorie tinies, Group1 e Group2.

I **tinies** contengono il conteggio di una determinata funzione, come mostrato in figura 1.1 per esempio vediamo il conteggio dei corsi erogati ecc.

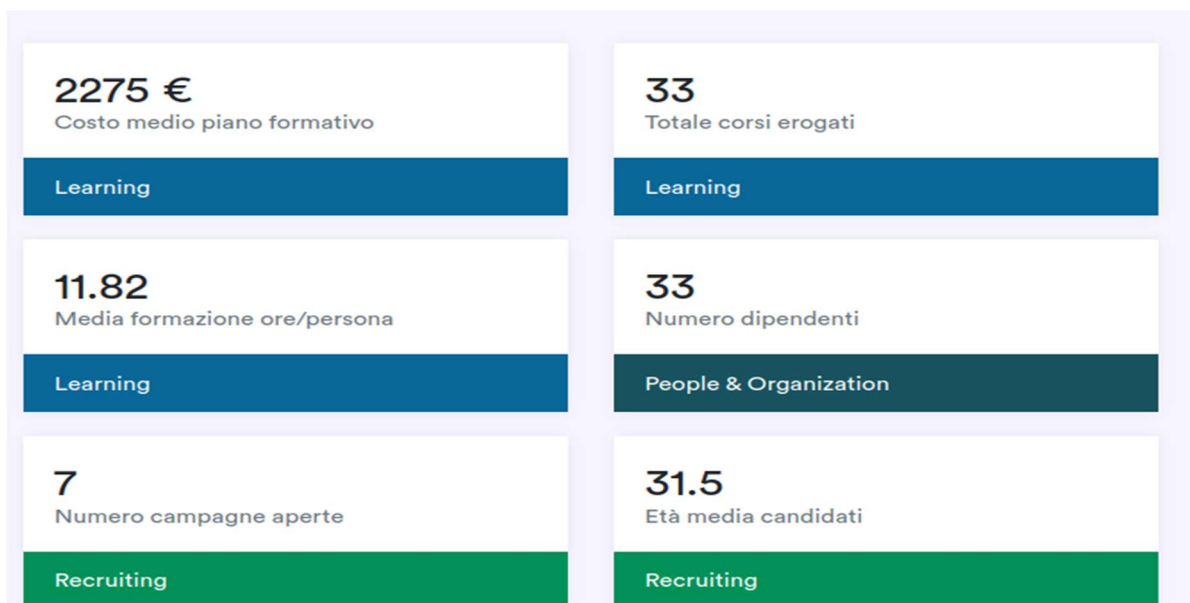


Figura 1.3.1 esempio di Tinies Widgets

Il Group1 contiene la rappresentazione dei dati sul grafico, come mostrato per esempio in figura 1.2.

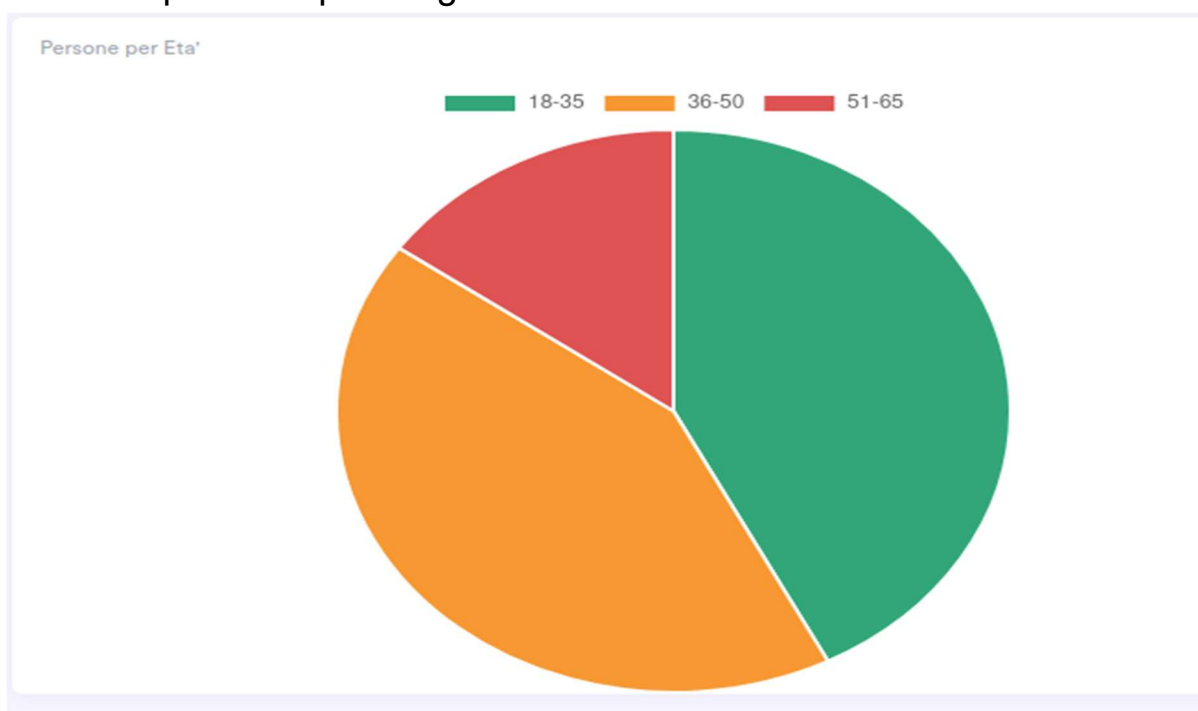


Figura 1.2 Esempio di Group1 Widgets

Il Group2 serve per rappresentare la quantità dei dati di una determinata specifica

Top 5 Skills più richieste

Nome	Quantità
Analisi dei flussi finanziari	1
Bilancio Civile, principi contabili e consolidamento bilancio	1
Capitalizzazioni e cespiti	1

Figura 1.3 Esempio di Group 2

Con questo widget è possibile gestire la documentazione aziendale, tenere traccia del numero di persone che hanno compilato le schede ecc. Inoltre il cliente può customizzare la dashboard scegliendo i widget che ritiene necessari.

## 1.4 – Scopo del Progetto

Lo scopo del progetto è la costruzione di un set di widget funzionali per fornire informazioni sui principali processi HR gestiti dalla piattaforma. I singoli widget saranno collegati ai moduli di cui si compone la piattaforma HRTTools, e andranno a comporre la dashboard di ciascun utente.

Il singolo widget sarà selezionabile da una lista in base al sistema di permessi collegato con la profilazione dell'utente. Ciascun widget avrà alcune features specifiche : utilizzo di filtri, possibilità di esportare in formato excel i dati di cui è composto, possibilità di essere stampato o esportato in pdf.

## Capitolo 2

### Tecnologie Utilizzate

#### 2.1 – Introduzione

In questo capitolo verranno descritte le scelte delle tecnologie utilizzate, che hanno preceduto la progettazione della base di dati e del sito web. In particolare, verrà fatto un confronto tra i principali framework per web App e DBMS. In questo capitolo saranno elencati gli strumenti di comunicazione utilizzati, insieme al metodo di sviluppo del software adottato e infine il framework.

#### 2.2 – Database

In informatica, con base di dati, banca dati, database o anche solo DB, si indica una collezione di dati correlati che, mediante un computer, vengono utilizzati per rappresentare una certa porzione del mondo reale. Un database è una collezione di dati strutturati e memorizzati all'interno di un sistema informatico. Lo scopo di un database è quello di raccogliere un certo insieme di dati per metterlo poi a disposizione di chi ne faccia richiesta, a prescindere se questo sia un semplice utente o un altro computer. In parole povere, dato che ormai le informazioni in formato elettronico sono sempre più numerose, un database serve sia per raccogliere queste informazioni che per metterle a disposizione degli altri.

Per quanto riguarda il database, laravel fornisce enormi vantaggi rispetto al semplice PHP e a molti altri framework, infatti permette di chiamare query direttamente al database come gli altri, ma possiede anche delle funzioni

proprie, legate alla ActiveRecord class, che permettono di fare richieste, update e insert in modo molto più semplificato senza nemmeno scrivere direttamente l'SQL.

Per dare la possibilità di creare, manipolare e consultare un database si utilizzano invece degli appropriati linguaggi di interrogazione attraverso dei particolari programmi detti comunemente DBMS (DataBase Management System). Per quel che riguarda la tipologia, un database in genere può essere o locale, cioè utilizzabile su un computer da uno o più utenti, o distribuito, ad indicare, cioè, che le informazioni in esso immagazzinate sono accessibili attraverso una rete informatica. I database vengono gestiti dai DBMS, software come Mysql workbench, Dbeaver che permettono la creazione e l'interrogazione dei dati.

MySQL è un DBMS di proprietà di Oracle, rilasciato sia su licenza GNU che commerciale. Essendo scritto in C++, offre un supporto multiplatforma vastissimo e molteplici linguaggi di programmazione ne permettono l'integrazione. Questo DBMS permette l'utilizzo di differenti Storage Engine, che permettono una buona versatilità e la possibilità di selezionare lo Storage Engine più appropriato per la gestione di determinate tipologie di database.

## 2.3 Applicazione Web

Con il termine “Applicazione Web” [3] si intende un'applicazione risiedente in un Server Web alla quale si accede tramite un browser Internet o un altro programma con funzioni di navigazione operante secondo gli standard del World Wide Web.

Per completezza va detto che con un termine simile, Servizi Web o Web Service, si intende un nuovo modo di realizzare applicazioni distribuite ad oggetti, in cui gli oggetti comunicano tra loro attraverso la rete tramite i protocolli SOAP (basato sullo standard XML) e HTTP.

Le applicazioni web non risiedono sulle macchine che le utilizzano, bensì

risiedono su server remoti in un'architettura client-server.

Le aziende come google, microsoft stanno facendo più web app possibile per dare ai utenti la scelta di non installare software non necessari.

Con la pandemia sviluppo dell'applicazione web è aumentato esponenzialmente per dare i servizi da remoto.

### 2.3.1 – Front-End

Con il termine Front-End ci si riferisce alla porzione della web-app visibile all'utente e con la quale un utente normale interagisce. L'interazione con l'applicazione avviene mediante una classica pagina web attraverso un browser, esso visualizza le informazioni e permette l'acquisizione di dati in input. Esistono molteplici linguaggi e framework con la quale è possibile realizzare i front-end. In questo progetto, si useranno il framework Vuejs e i seguenti linguaggi :

- HTML (HyperText Markup Language): utilizzato per definire la struttura delle pagine web

- CSS (Cascading Style Sheets): utilizzato per definire la struttura e la formattazione grafica delle pagine web

- JavaScript: linguaggio che permette di rendere dinamiche alcune sezioni della pagina, mediante l'esecuzione di script da parte del browser; mediante questo linguaggio verrà validata localmente parte dei dati in input, prima di inviarli al server

- AJAX (Asynchronous Javascript and XML): utilizzato per la gestione di alcuni contenuti dinamici delle pagine

- jQuery: libreria Javascript che semplifica la gestione dei contenuti dinamici delle pagine web, mediante una gestione modulare degli eventi della pagina

- Bootstrap: è un framework CSS gratuito e open source diretto allo sviluppo web front-end reattivo e mobile-first. Contiene modelli di progettazione basati su CSS e (facoltativamente) JavaScript per tipografia, moduli, pulsanti, navigazione e altri componenti dell'interfaccia

- Blade: è il motore di creazione di modelli semplice ma potente incluso in Laravel. A differenza di alcuni motori di template PHP, Blade non ti impedisce di utilizzare il semplice codice PHP nei tuoi template. In effetti, tutti i modelli Blade vengono compilati in semplice codice PHP e memorizzati nella cache fino a quando non vengono modificati, il che significa che Blade aggiunge essenzialmente zero overhead alla tua applicazione.

- Vue js (pronunciato / vju: /, come view): è un framework progressivo per la creazione di interfacce utente. A differenza di altri framework monolitici, Vue è progettato da zero per essere adottabile in modo incrementale. La libreria principale è focalizzata solo sul livello di visualizzazione ed è facile da raccogliere e integrare con altre librerie o progetti esistenti.

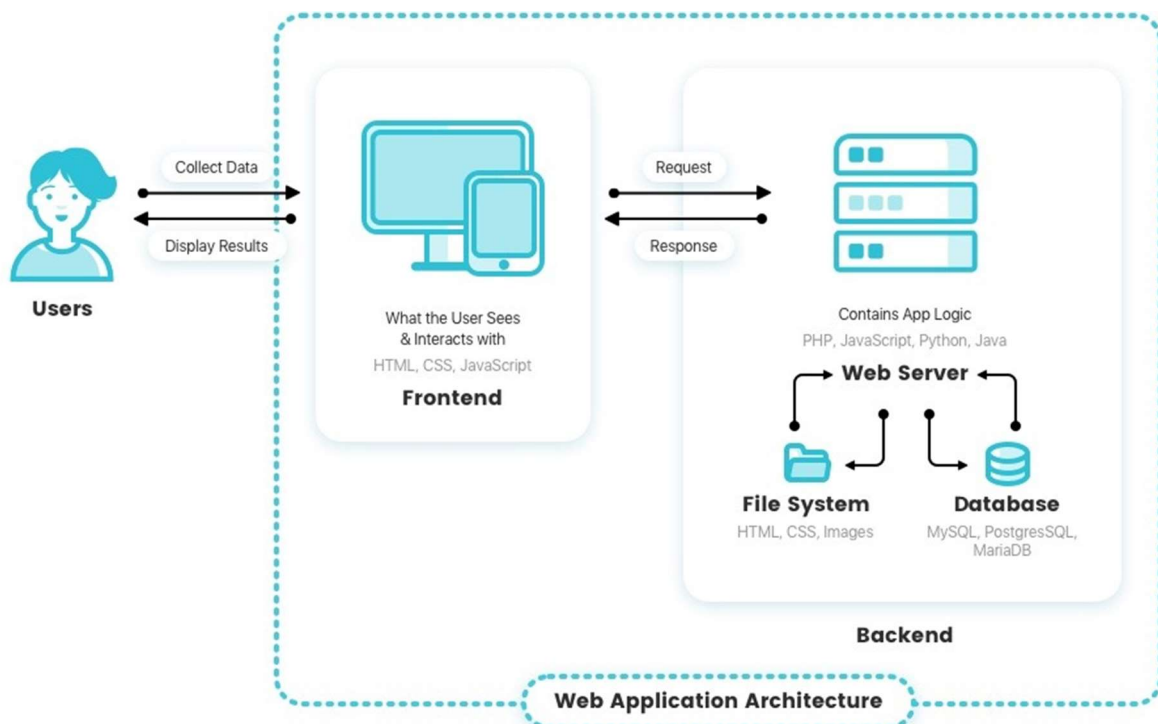


Figura 2.1 Diagramma Architeturale di Applicazione web

### 2.3.2 – Back-End

Con il termine Back-End si indica la porzione della web-app che si occupa dell'interazione tra l'utente ed i dati memorizzati all'interno del database. Questa parte dell'applicazione gestisce : l'autenticazione dell'utente, get, store or update di dati presi dal front end.



Il back-end verrà realizzato principalmente mediante l'utilizzo di PHP, unito ad HTML, CSS, e JavaScript che costituiranno il corpo delle pagine inviate al front-end. In questo caso la parte di Back-End è basata su framework Laravel.

## 2.4 – Ambienti di Sviluppo

Per poter semplificare il lavoro e risparmiare il tempo con dei shortcut è stato necessario l'utilizzo di ambienti di sviluppo integrati (IDE).

- PhpStorm: IDE specializzato per lo sviluppo del back-end dei siti web mediante il linguaggio PHP, Integrazione dei sistemi di controllo Gitlab che consente di eseguire comandi (commit, merge, diff, ecc.) direttamente dall'IDE.

- Dbeaver: DBeaver è uno strumento di database universale gratuito e open source per sviluppatori e amministratori di database.

## 2.5 – Strumenti di Comunicazione

La realizzazione di questo progetto ha richiesto l'utilizzo di svariati strumenti di comunicazione come Skype, Gitlab e Asana.

**Skype:** La comunicazione con l'azienda è avvenuta tramite Skype, il smartworking non è una cosa nuova per emme delta, infatti molti dei lavoratori lavoravano già da remoto prima del SARS COVID-19.

**Gitlab:** Gitlab è una piattaforma web open source che permette la gestione di repository Git e di funzioni trouble ticket. Ogni membro del team all'interno di Git poteva lavorare nel suo branch (la 'cratella' dove viene caricato il lavoro) e a fine giornata si fa il merge con il branch master dopo aver testato il codice.

**Asana:** Asana è tra le web application da prendere in considerazione per organizzare le attività in team di lavoro sempre più orientati allo smart working. Per ottimizzare il tempo sul lavoro occorre quindi dotarsi degli

strumenti giusti e i software di collaborazione come Asana o Trello che possono rivelarsi dei validi alleati.

## 2.6 – Metodo di Sviluppo Agile

La situazione di emergenza sanitaria mi ha portato allo sviluppo del progetto a distanza ed in maniera abbastanza autonoma. Per questo motivo l'azienda ha optato per metodo di sviluppo agile.

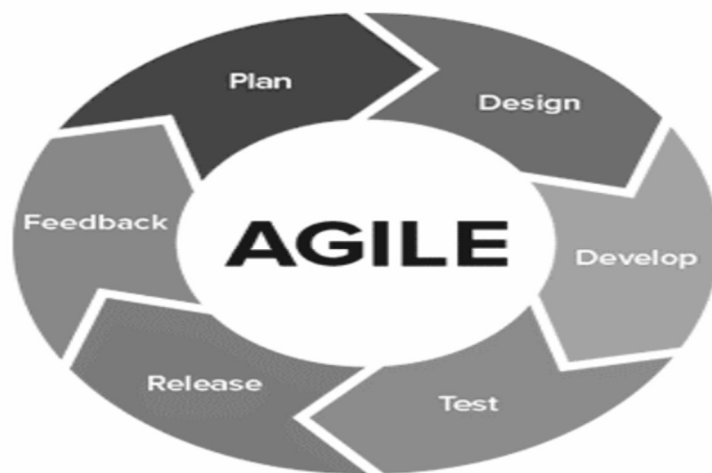


Figura 2.2 Diagramma Metodo Agile

La "metodologia agile" [2] è un approccio allo sviluppo del software basato sulla distribuzione continua di software efficienti creati in modo rapido e iterativo. Benché l'espressione faccia riferimento ad un determinato sviluppo del software questa metodologia non prevede regole da rispettare tassativamente nell'ambito dello sviluppo del software. Si tratta piuttosto di un tipo di approccio alla collaborazione e ai flussi di lavoro fondato su una serie di valori in grado di guidare il nostro modo di procedere.

In pratica, le metodologie di sviluppo software agile consistono nel rilasciare rapidamente modifiche al software in piccole porzioni con l'obiettivo di migliorare la soddisfazione dei clienti.

## Capitolo 3

### Analisi di Laravel

#### 3.1 - Introduzione

In questo capitolo verranno descritte il famoso framework laravel, che tipo di architettura uso e tutti suoi vantaggi. hanno preceduto la progettazione della base di dati e del sito web.

Laravel è un framework open source e ad alte prestazioni per la realizzazione di applicazioni web. Il suo fine è quello di alleggerire il lavoro legato allo sviluppo delle attività più semplici di un'applicazione web da parte dello sviluppatore.(Laravel alleggerisce molto il lavoro di sviluppatori di un'applicazione web e rende molto semplice da lavorare ). Il framework fornisce delle librerie per la base di dati, la creazione di template in HTML e Blade(template engine).

Laravel gestisce la sessione dell'utente con dei semplici comandi. (Il Framework fornisce delle librerie per creare la parte di registrazione e login e inoltre crea anche i modelli per base di dati per utente). Ma soprattutto permette di riuso del codice che permette così anche di estendere codici di siti già scritti precedentemente e rendendo molto modulare e flessibile. Secondo la classifica di gitlab il Laravel è il framework più utilizzato nel 2020, Nelle grandi organizzazioni come banche e grandi aziende viene spesso utilizzato questo framework proprietario per aumentare il livello di sicurezza. Laravel è stato scritto in 2011 per versioni di Php dalla 5.7 in avanti.

Laravel in confronto con altri framework:

Come quasi tutti i framework PHP, Laravel implementa il pattern architetturale MVC(Model- View-Controller) e organizza il codice sulla base di quel modello. Laravel segue la filosofia che il codice deve essere scritto in modo semplice e elegante.

Laravel è un framework full-stack che fornisce molte funzionalità collaudate e pronte per essere utilizzate come Dump and Die(Uno dei modi più comuni per eseguire il debug), Laravel Tinker (ti consente di interagire con un database senza creare le rotte).

Quando si crea un'applicazione web, sono disponibili numerosi strumenti e framework. Tuttavia, riteniamo che Laravel sia la scelta migliore per la creazione di applicazioni web moderne e full-stack. Laravel combina i migliori pacchetti nell'ecosistema PHP per offrire il framework più robusto e intuitivo per gli sviluppatori disponibile. Inoltre, migliaia di sviluppatori di talento da tutto il mondo hanno contribuito al framework.

### 3.2 - Pattern MVC

Come detto in precedenza, Laravel predilige il pattern architetturale MVC(Model-View-Controller). Laravel implementa artisan, un code generator molto utile ed efficiente che permette al programmatore di risparmiare tempo nella scrittura del codice. Può creare con un semplice comando da terminale un modello con le tabelle del database e il controller che contiene le funzionalita' : index, create, edit, show, delete. Le colonne della tabella selezionata andranno a definire gli attributi del model, mantenendo i tipi di dato che sono stati definiti nel database.

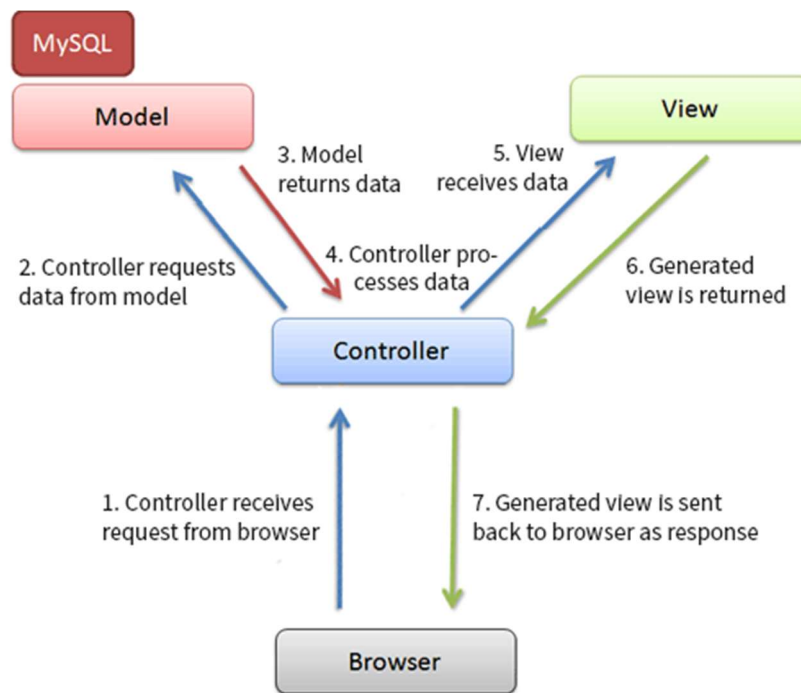


Figura 3.1 Diagramma Model View Controller di Laravel

Models: creando il nostro primo modello

Creiamo il nostro primo modello, M in MVC, per la nostra applicazione. Come abbiamo notato prima, il modello di solito si interfaccia con un archivio dati come un database SQL. In Laravel, il modello è solitamente una classe con proprietà che corrispondono alle colonne del database.

Nel nostro database, un prodotto avrà le seguenti proprietà:

- Name (name) – Nome di una persona.
- Surname(surname) - Cognome di una persona
- Email(email)- Email di una persona

Per creare un modello in Laravel, esegui il comando nel tuo terminale:

```
$ php artisan make:model Persona
```

Quando esegui questo comando, Laravel creerà un file Persona.php nella directory dell'app/Models. Questa sarà una classe PHP con il nome Persona e sarà il modello per la nostra tabella dei prodotti nel database.

La migrazione consente agli sviluppatori di apportare e annullare le modifiche al database di un progetto. Le migrazioni possono essere utilizzate per rendere la gestione dei database facile e prevedibile. Per creare una migrazione, basta eseguire il seguente comando nel terminale:

```
$ php artisan make:migration create_User_table
```

Quando il comando viene eseguito, dovremmo vedere un nuovo file : `_create_user_table.php` nella cartella database del progetto e possiamo modificarlo per avere lo schema della tabella dei nostri prodotti in questo modo:

```
<?php

class CreateUserTable extends Migration
{
    public function up()
    {
        Schema::create(user, function (Blueprint $table) {
            $table->increments('id');

            $table->string('name');

            $table->string('surname');

            $table->string('email')->unique();

            $table->softDeletes();

            $table->timestamps();

        });
    }
}
```

## Controller: creazione del nostro controller

In precedenza abbiamo accennato al fatto che i controller sono responsabili del completamento delle azioni dell'utente e della gestione della logica di business delle nostre applicazioni. Per il nostro progetto fittizio, utilizzeremo i controller delle risorse

Per creare un controller di risorse in Laravel si fa attraverso il seguente comando:

```
$ php artisan make:controller UserController -r
```

Il flag -r lo rende un controller di risorse e quindi crea tutti i metodi richiesti per l'operazione CRUD(show, create, edit, update, delete). Quando il comando viene eseguito, Laravel creerà un nuovo file UserController.php nella cartella app/Http/Controllers.

Invece di definire tutta la logica di gestione delle richieste come chiusure nei file di route, potresti voler organizzare questo comportamento utilizzando le classi "controller". I controller possono raggruppare la logica di gestione delle richieste correlate in una singola classe.

Diamo un'occhiata a un esempio di controller di base. Si noti che il controller estende la classe controller di base inclusa con Laravel:

```
App\Http\Controllers\Controller:
```

```

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\Models\User;

class UserController extends Controller
{
    /**
     * Show the profile for a given user.
     *
     * @param int $id
     * @return \Illuminate\View\View
     */
    public function show($id)
    {
        return view('user.profile', [
            'user' => User::findOrFail($id)
        ]);
    }
}

```

Si può definire un percorso per questo metodo del controller in questo modo:

```

use App\Http\Controllers\UserController;

Route::get('/user/{id}', [UserController::class, 'show']);

```

Quando una richiesta in arrivo corrisponde all'URI della route specificato,



verrà richiamato il metodo show nella classe App \ Http \ Controllers \ UserController e al metodo verranno passati i parametri della route.

## Resource Controllers

Se si pensa a ogni modello Eloquent nella propria applicazione come a una "risorsa", è tipico eseguire gli stessi set di azioni su ciascuna risorsa nella propria applicazione. Ad esempio, immagina che la tua applicazione contenga un modello di foto e un modello di film. È probabile che gli utenti possano creare, leggere, aggiornare o eliminare queste risorse.

Azioni gestite da Resource Controller:-

Verb	URI	Action	Route Name
GET	/photos	index	photos.index
GET	/photos/create	create	photos.create
POST	/photos	store	photos.store
GET	/photos/{photo}	show	photos.show
GET	/photos/{photo}/edit	edit	photos.edit
PUT/PATCH	/photos/{photo}	update	photos.update
DELETE	/photos/{photo}	destroy	photos.destroy

Views: creazione delle views(viste) dei progetti

In Laravel, tutte le viste sono archiviate nella directory resources / views. Le viste di solito memorizzano l'HTML della tua pagina e sono il livello di presentazione dell'architettura MVC.

Puoi creare una vista inserendo un file con estensione blade.php nella directory delle resources / views della tua applicazione. L'estensione blade.php informa il framework che il file contiene un template Blade. Laravel utilizza Blade come motore di creazione dei modelli. Blade è praticamente

HTML ma con una sintassi simile a PHP.

I modelli Blade contengono HTML e direttive Blade che consentono di riprodurre facilmente i valori, creare istruzioni "if", iterate, while sui dati e altro ancora.

Questa visualizzazione sopra è un semplice modulo che raccoglie e invia richieste per creare prodotti. Quando il modulo viene inviato, viene effettuata una richiesta POST al percorso /user dell'applicazione che viene gestita dal metodo store nel nostro UserController.php.

Possiamo passare un array di dati alle view per rendere disponibili alla view:

```
return view(fileName, ['ObjKey' => 'ObjValue']);
```

Conclusione: In questo capitolo, abbiamo considerato come funziona MVC e come Laravel lo implementa. Abbiamo considerato il motivo per cui bisognerebbe usare MVC e come implementarlo in un'applicazione Laravel del mondo reale.

### 3.3 - Database

Per quanto riguarda il database laravel fornisce enormi vantaggi rispetto al semplice PHP e a molti altri framework, infatti permette di chiamare query dirette al database come gli altri, ma possiede anche delle funzioni proprie, legate alla ActiveRecord class, che permettono di fare richieste, update e insert in modo molto più semplificato senza nemmeno scrivere direttamente l' SQL.

La configurazione per i servizi di database di Laravel si trova nel file di configurazione config /database.php dell' applicazione. In questo file è possibile definire tutte le connessioni al database, nonché specificare quale connessione deve essere utilizzata per impostazione predefinita. La maggior parte delle opzioni di configurazione all'interno di questo file sono guidate dai valori delle variabili di ambiente dell'applicazione. In questo file vengono forniti esempi per la maggior parte dei sistemi di database supportati da Laravel.

Php artisan make: migration create\_User\_table

Con questo andiamo a creare un file che ci servirà per creare un table dentro il database, una volta eseguito il comando apriamo un nuovo file per inserire i dati.

Php artisan migrate : con questo comando andiamo inserire i dati nel database e creare nuove tabelle o modificare le tabelle.

Una volta inseriti i dati si può annullare la migrazione con il seguente comando : Php artisan rollback

### 3.4 - Seeding

Laravel include la possibilità di eseguire il seeding del database con dati di test utilizzando le classi seed. Tutte le classi seed vengono memorizzate nella directory database / seeder.

Per generare un seeder, eseguire il comando make: seeder Artisan. Tutti i seeders generati dal framework verranno inseriti nella directory database / seeders:

php artisan make:seeder UserSeeder

Ad esempio, modifichiamo la classe Database Seeder predefinita e aggiungiamo un'istruzione di inserimento del database al metodo run:

```

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;

class DatabaseSeeder extends Seeder
{
    /**
     * Run the database seeders.
     *
     * @return void
     */
    public function run()
    {
        DB::table('users')->insert([
            'name' => Str::random(10),
            'email' => Str::random(10).'@gmail.com',
            'password' => Hash::make('password'),
        ]);
    }
}

```

## Parte II

### Progetto e Sviluppo

# Capitolo 4

## Progettazione

### 4.1 - Introduzione

Questo capitolo ha l'obiettivo di descrivere le fasi per la progettazione del modulo di valutazione e la progettazione dei widget che consente di verificare i numeri di schede che sono state completamente compilate, quella compilate parzialmente e quelle già archiviate. I widget sono box e grafici in grado di fornire dati e informazioni legati a tutti i processi gestiti in HRTTools.

La prima parte di progettazione è dedicata all'analisi dei requisiti del cliente descritti da Emme Delta.

La parte successiva riguarda la progettazione della database che verrà dimostrato successivamente con lo schema ER[4].

Nell'ultima parte verranno mostrati i widget, la loro interfaccia e la gestione del caricamento dei dati all'interno del database.

### 4.2 - Requisiti funzionali

L'analisi dei requisiti costituisce una delle fasi più importanti per la realizzazione di un progetto. Il fine di questo processo è quello di stabilire quali sono i requisiti del cliente e quali sono i vincoli da rispettare. E' importante renderli chiari sia per il team di sviluppo che per i clienti. In genere, i requisiti funzionali descrivono il comportamento del sistema in condizioni specifiche.

L'elenco di esempi di requisiti funzionali include:

- Studio di fattibilità
- Correzioni, aggiustamenti e cancellazioni delle transazioni
- Funzioni amministrative
- Autenticazione

I requisiti si dividono in tre tipologie:

- requisiti funzionali: descrivono cosa avrà da offrire il sistema e come dovrà reagire agli input
- requisiti non funzionali: descrivono le caratteristiche dei servizi offerti dal sistema, come per esempio l'affidabilità, la velocità di elaborazione, ecc.
- requisiti di dominio: derivano immediatamente dal dominio dell'applicazione

#### 4.2.1 – Requisiti di Base

Si vuole progettare una sistema di valutazione di personali per consentire di pianificare la politica retributiva dell'azienda e alcuni widgets per analizzare i dati.

Emma delta, come progetto del mio tirocinio, mi ha commissionato la realizzazione di un modulo behaviors partendo da una base di dati.

Una volta realizzato il database il mio compito consisteva nell' implementarlo su HRTools e i contenuti memorizzati nel database dovevano essere accessibili mediante un'applicazione web con la possibilità di esportazione di dati tramite un pulsante. L'applicazione e la base di dati dovranno essere compatibili con gli altri moduli di sistema preesistente.

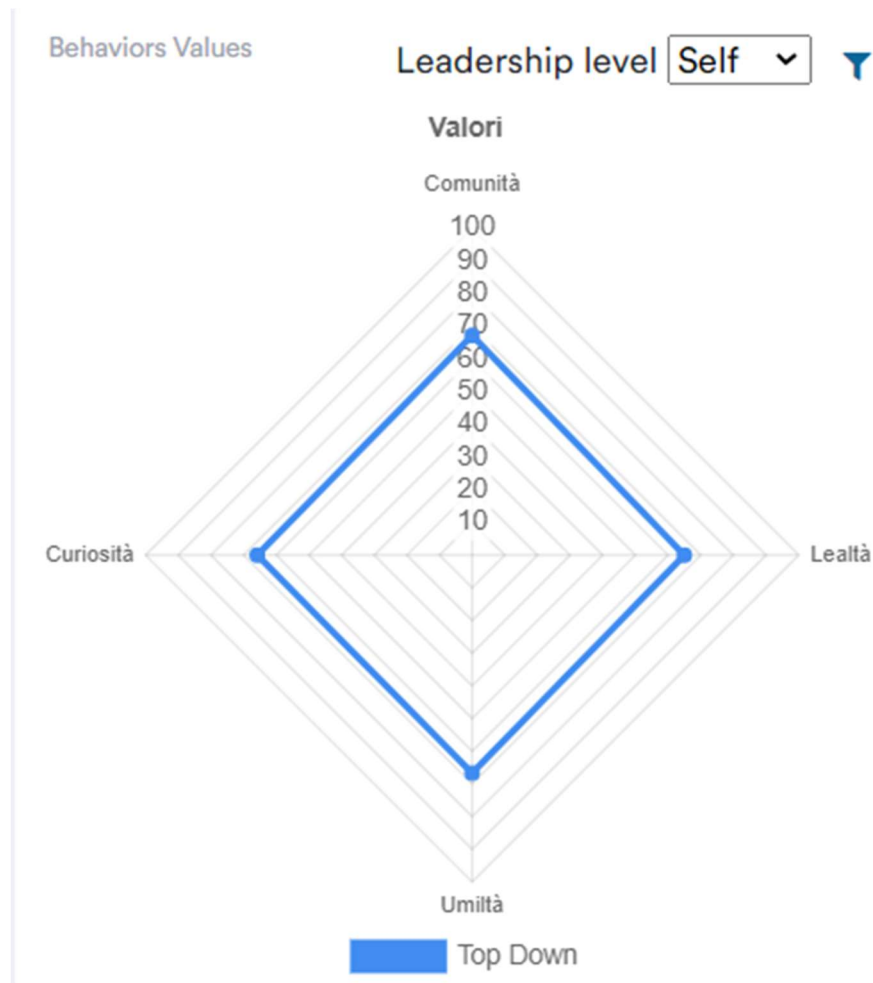


Figura 4.1 Esempio di un widget di comportamento.

#### 4.2.2 – Requisiti Aggiuntivi

Il più importante dei requisiti aggiuntivi, che emme delta ha specificato durante lo svolgimento del mio tirocinio, è stata la realizzazione un modulo che permette di esportare i dati mettendo una query dall'interfaccia con alcuni parametri come per esempio inizio data, fine data.

#### 4.3 - Schema ER di Behavior Set

Il database per behaviour era fatto con le seguenti tabelle.

Behaviour set serve per selezionare se la valutazione veniva da un team o



da un responsabile.

Behaviour skills per identificare il tipo di abilità da valutare.

Behaviour indicators per una breve descrizione del comportamento.

Behaviour values per selezionare varie tipologie di valori.

### Schema Scheletro

La gestione dell'questionario dei moduli Behaviours richiede la memorizzazione delle relative tabelle di behaviours set, behaviours skills, behaviours values. Viene di seguito riportato lo schema scheletro relativo a questa porzione del database (Figura 4.2)

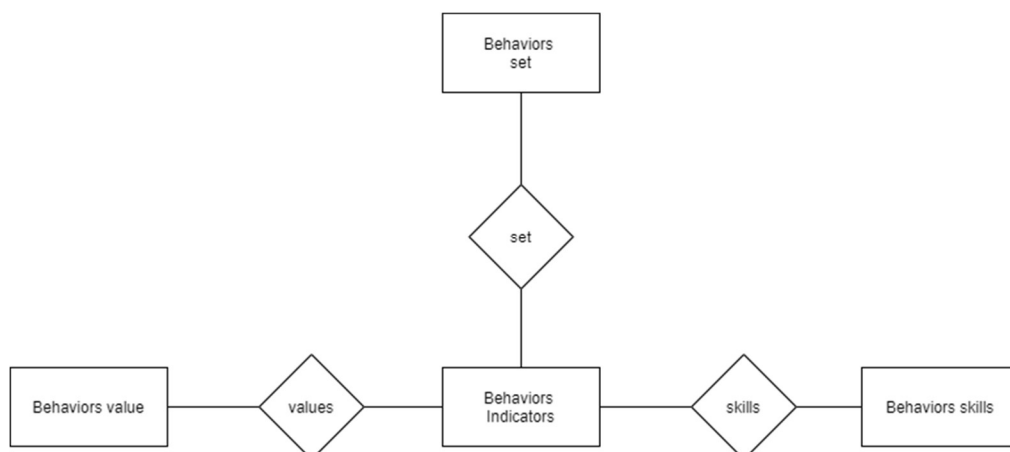


Figura 4.2 Schema scheletro di Modulo Behaviours

## Attributi e Chiavi

Nella seguente figura (Figura 4.3) sono riportati gli attributi delle varie entità.

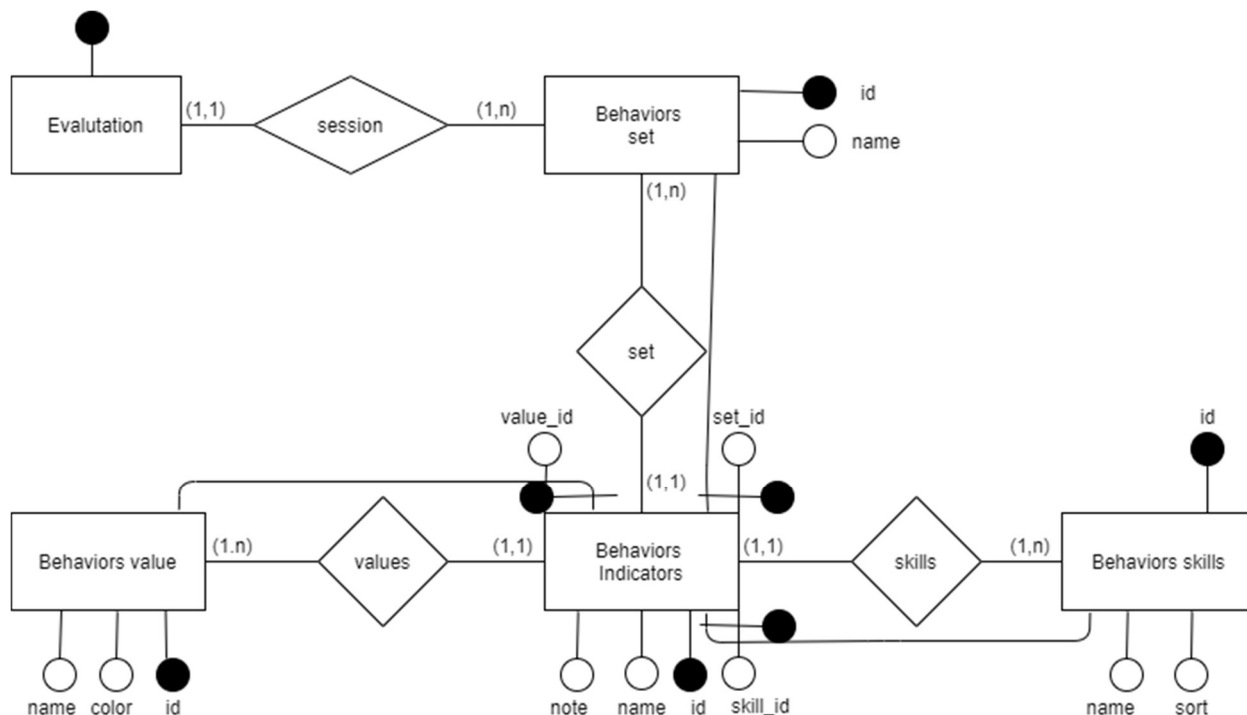


Figura 4.3 Schema ER di Modulo Behaviours

### 4.3.1 – Gestione Utenti

#### Schema Scheletro

La gestione degli utenti, identifica tre tipologie di utenti: Amministratore, Responsabile e Utente normale(Workers). L'applicativo prevede un sistema di permessi che consente agli utenti admin di avere accesso a tutte le funzionalità offerte e a tutti i dati immagazzinati, mentre agli utenti “workers” è consentito accedere solo ad alcune parti. Viene di seguito riportato lo schema scheletro relativo a questa porzione del database.

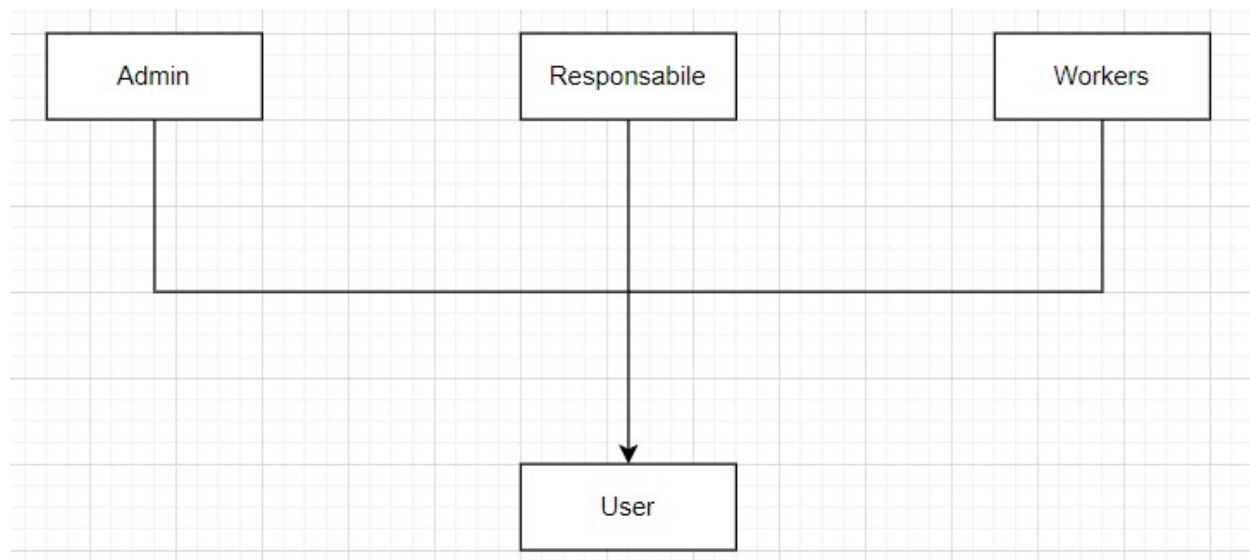


Figura 4.4 – Schema Scheletro Gestione Utenti

### Attributi e Chiavi

La gestione degli utenti vede la presenza di una generalizzazione. Tale generalizzazione risulta essere di tipo totale ed esclusiva (in quanto le tipologie di utenti sono distinte – un Responsabile non è anche amministratore).

Nella seguente figura (Figura 4.5) sono riportati gli attributi e le chiavi delle entità. Gli attributi delle entità figlie sono semplicemente quelli ereditati dall'entità padre.

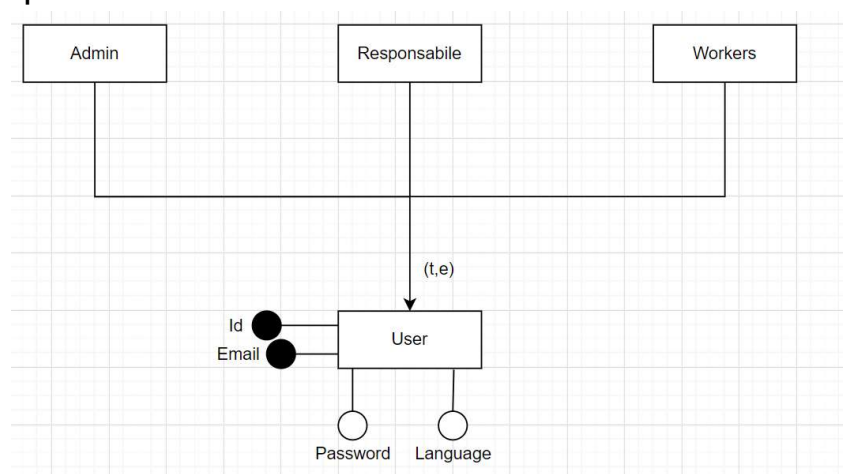


Figura 4.5 – Diagramma ER Gestione Utenti: Attributi e Chiavi

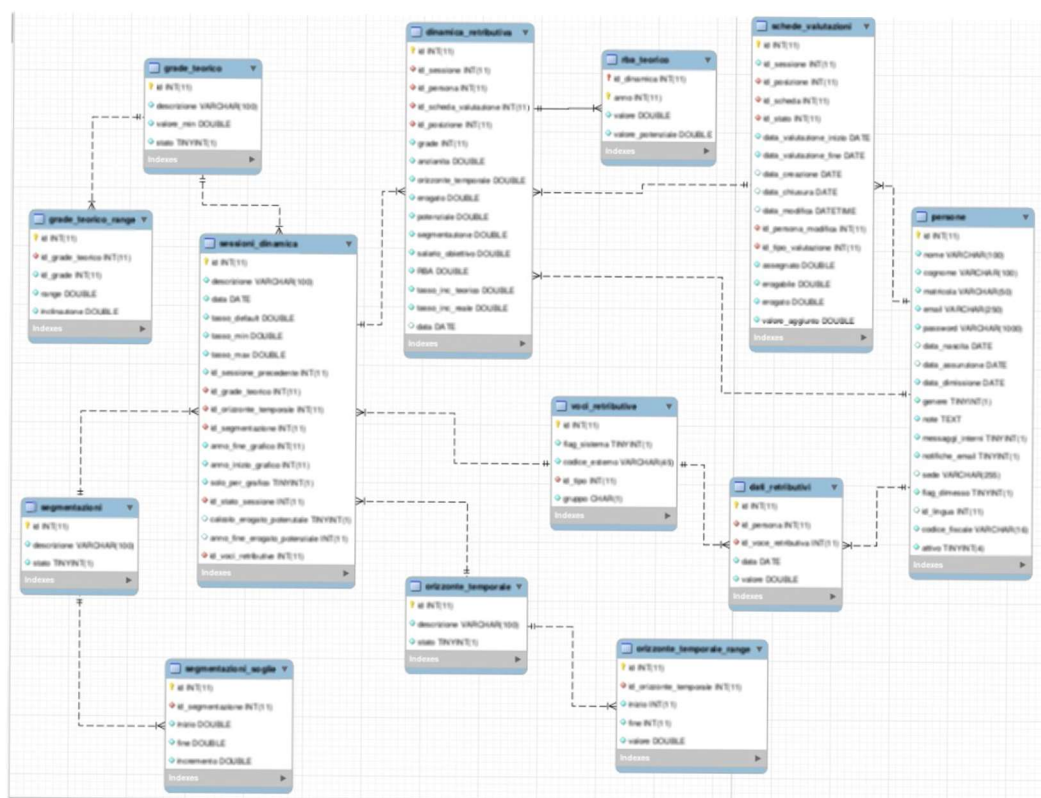


Figura 4.6 – Schema Fisico di base di dati

Tutto il codice, che è stato sviluppato durante il lavoro di tirocinio, gira intorno a queste tabelle. Le colonne con una chiave gialla sono le primary key, quelle con rombo rosso sono le foreign key.

Le altre colonne azzurre sono required e quindi not null, mentre se il rombo è di colore bianco possono anche assumere il valore null.

### 4.3 - Progettazione dei widgets per valutazione

In questa parte verrà analizzata la progettazione del modulo di valutazione che consente di pianificare la politica retributiva delle aziende e i suoi widgets. Per dare una visione generale su come si presenta l'ambiente di lavoro di HRTools viene mostrata di seguito la schermata di interfaccia del modulo di valutazione:

Seleziona una sessione

Sessione 2C00 (7/10/2021 - 14/10/2021)

▼

Reset Personal Skills

Download Xlsx

All

▼

Cerca...

🔍







Valutato		Valutatore	Ruolo	Stato	Tipo Valutazione
<input type="checkbox"/> Sacconi Claudio <div>Roles</div>		Presidente Sezione	Collaboratore Content	 Chiusa (Autovalutazione)	self
<input type="checkbox"/> Sacconi Claudio <div>Roles</div>		Presidente Sezione	Collaboratore Content	 Colloquio (Top-Down)	top_down
<input type="checkbox"/> Berta Matteo <div>Roles</div>		Berta Matteo	Collaboratore Logistica	 Chiusa (Autovalutazione)	self
<input type="checkbox"/> Berta Matteo <div>Roles</div>		Presidente Sezione	Collaboratore Logistica	 Colloquio (Top-Down)	top_down
<input type="checkbox"/> Elio Di Gennaro <div>Roles</div>		Elio Di Gennaro	Collaboratore - Comunicazione Aziendale	 Chiusa (Autovalutazione)	self
<input type="checkbox"/> Elio Di Gennaro <div>Roles</div>		Presidente Sezione	Collaboratore - Comunicazione Aziendale	 Colloquio (Top-Down)	top_down

Figura 4.7 – Esempio schede di Valutazione

La parte del front end è stata realizzata con html e vuejs che consente di creare le parte di codice utilizzabile anche per altri moduli per esempio la rappresentazione di colonne o di ricerca che è stata implementata anche in altri moduli variando i parametri.

Per realizzare la parte di valutazione è stato necessario creare un modulo di behaviour(comportamento).

Behaviours   Indicatori		
Seleziona una Set		
Team		
<div>10</div> <div>All</div> <div>Cerca...</div>		
< <input type="checkbox"/> Indicator	Skill	Value
<input type="checkbox"/> Promuove ed è di esempio per i valori aziendali	Consistenza	Comunità
<input type="checkbox"/> Trasmette passione ed entusiasmo	Consistenza	Curiosità
<input type="checkbox"/> Si assume le responsabilità e condivide i successi con il proprio team	Consistenza	Lealtà
<input type="checkbox"/> Si mette in discussione	Consistenza	Umità
<input type="checkbox"/> È comprensivo con le necessità personali	Intelligenza Emotiva	Comunità
<input type="checkbox"/> È disponibile verso le richieste dei collaboratori	Intelligenza Emotiva	Curiosità
<input type="checkbox"/> Si mette nei panni dei collaboratori	Intelligenza Emotiva	Lealtà

Figura 4.8 - Esempio di un Behaviour set

Una volta creati i modelli e database per behaviour, e poi sono state create le parte di logica(Controllers) e di grafica(Views).

Per la parte di logica ho seguito la schema fornita dal resource di laravel che consente di scrivere il codice in poco tempo nei apposti controller, Per la parte grafica ho creato una cartella per ogni Behaviours (skill, indicator e value) e dentro ogni cartella ho creato dei files new, edit e index.

Il file new serve per aggiungere un nuovo elemento nel database e edit serve per modificare, mentre la index serve per visualizzare tutti gli elementi.

Il file index fa uso di vuejs per creare delle tabelle uniformi(come quelle degli altri modelli) dove di solito la prima colonna rappresenta il nome dell'elemento, la seconda serve per le actions (edit, delete, ecc) che puo essere customizzata con vue js.

Quando ho terminato la parte di Behaviour è stata implementata su delle

schede di valutazione. La valutazione delle competenze, l'assegnazione ed il monitoraggio di obiettivi sono rese possibili attraverso la costruzione di schede di valutazione personalizzabili e collegate con gli altri moduli di HRTools.

Un esempio di implementazione su una scheda di valutazione.

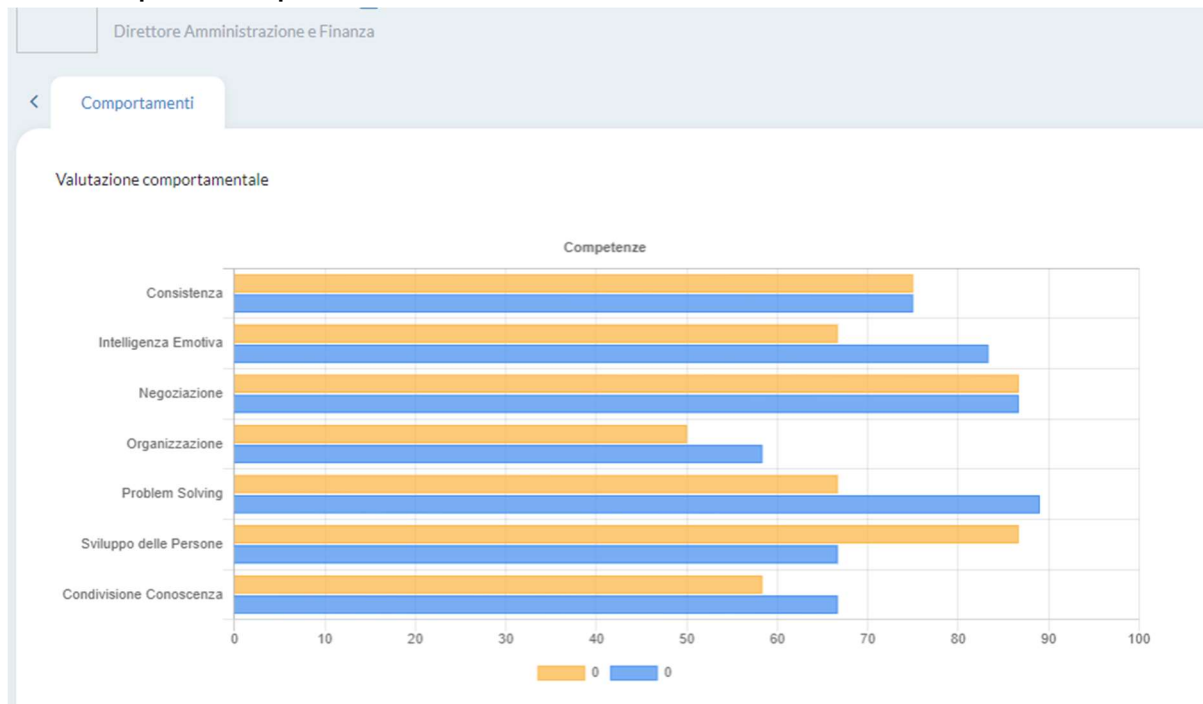


Figura 4.9 – Esempio di Behaviour Values

Il modulo scheda di valutazione offre funzionalità più importanti che danno una completa visione sulle caratteristiche del candidato attraverso vari dati di Behaviours. Come si può vedere nella figura 4.6 vengono evidenziati alcuni valori di Behaviours skills. Attingendo ai dati inseriti nei moduli di comportamento descritti precedentemente, consente di impostare delle sessioni di valutazione allo scopo di misurare l'adeguatezza dei dipendenti al ruolo ricoperto. Inoltre sulla base delle competenze e delle conoscenze, è possibile scegliere di valutare le prestazioni dei dipendenti in funzione di fattori differenti e personalizzabili da parte degli utenti.

Le schede offrono la possibilità di mostrare dei widgets che sintetizzano i dati dal database. Il modulo di valutazione prevede che ciascuna scheda,

una volta creata, sia compilata da un valutatore, revisionata da un responsabile gerarchico e infine inviata al valutato per essere approvata definitivamente.

Il vincolo aggiuntivo richiesto da uno dei clienti era di creare anche un widget che permette di controllare il numero di schede compilate parzialmente, il numero di anche quelle compilate completamente e il numero di quelle già archiviate. Ho scelto di mostrare queste informazioni su un grafico a torta come vediamo nella figura 4.7.

Una volta scelta la scheda, controllo per ogni utente se ha risposto a tutte le domande. Se si controllo se la sua scheda è stata già archiviata. Dopo aver attutao questo controllo aumentavo il mio contatore(numero di schede), poi i dati venivano inviati in oggetti in json per essere rappresentati su un widget.

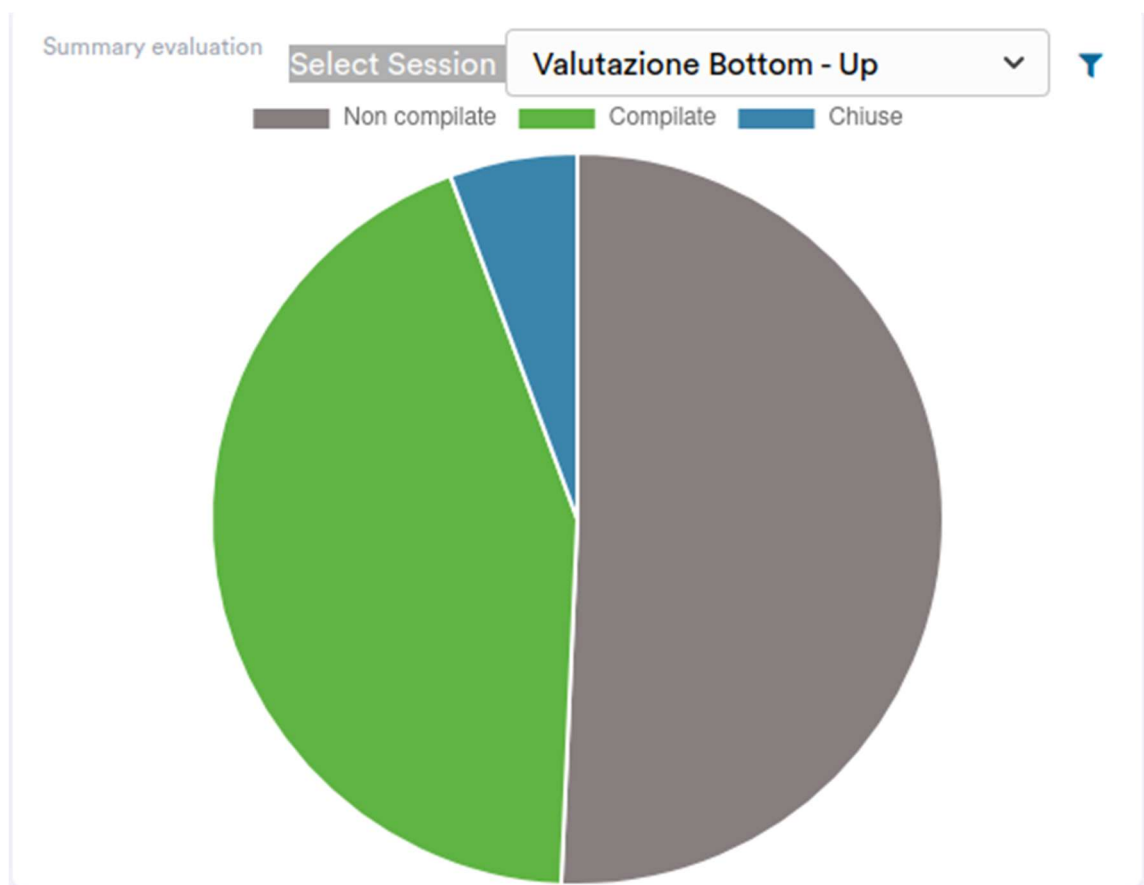


Figura 4.10 Esempio di un widget di controlla schede compilate



## 4.4 - Interfaccia

Interfaccia ha una struttura a blocchi divisa in due distinte sezioni tipiche della pagina di HTML: la sidebar, il body. La sidebar è impostata nella parte di codice del layout, che quindi viene definita in modo standard per tutto il sistema, mentre sarà il body(corpo della pagina) a cambiare in base al modulo del sito in cui ci si trova. Nella sidebar viene impostato un logo del HRTTools, una foto di profilo dell'utente e infine un alert per le notifiche(ché ci tiene aggiornati sulle novità').

Ogni dipendente può cambiare la sua foto tramite una libreria che consente di tagliare l'immagine per adattarla alla dimensione. Subito dopo c'è il pulsante dashboard che porta alla schermata che visualizza tutti i widget riguardanti l'utente loggato in base al gruppo a cui appartiene e al modulo selezionato.

Un admin o responsabile può modificare i permessi dal modulo che si occupa dell'anagrafica per degli utenti senza privilegi, i permessi che possono essere assegnati ad un utente sono per esempio: il permesso di guardare un modulo o di modificarlo. Questa parte è stata implementata dal middleware di laravel. Il middleware fornisce un comodo meccanismo per ispezionare e filtrare le richieste HTTP che entrano nell'applicazione.

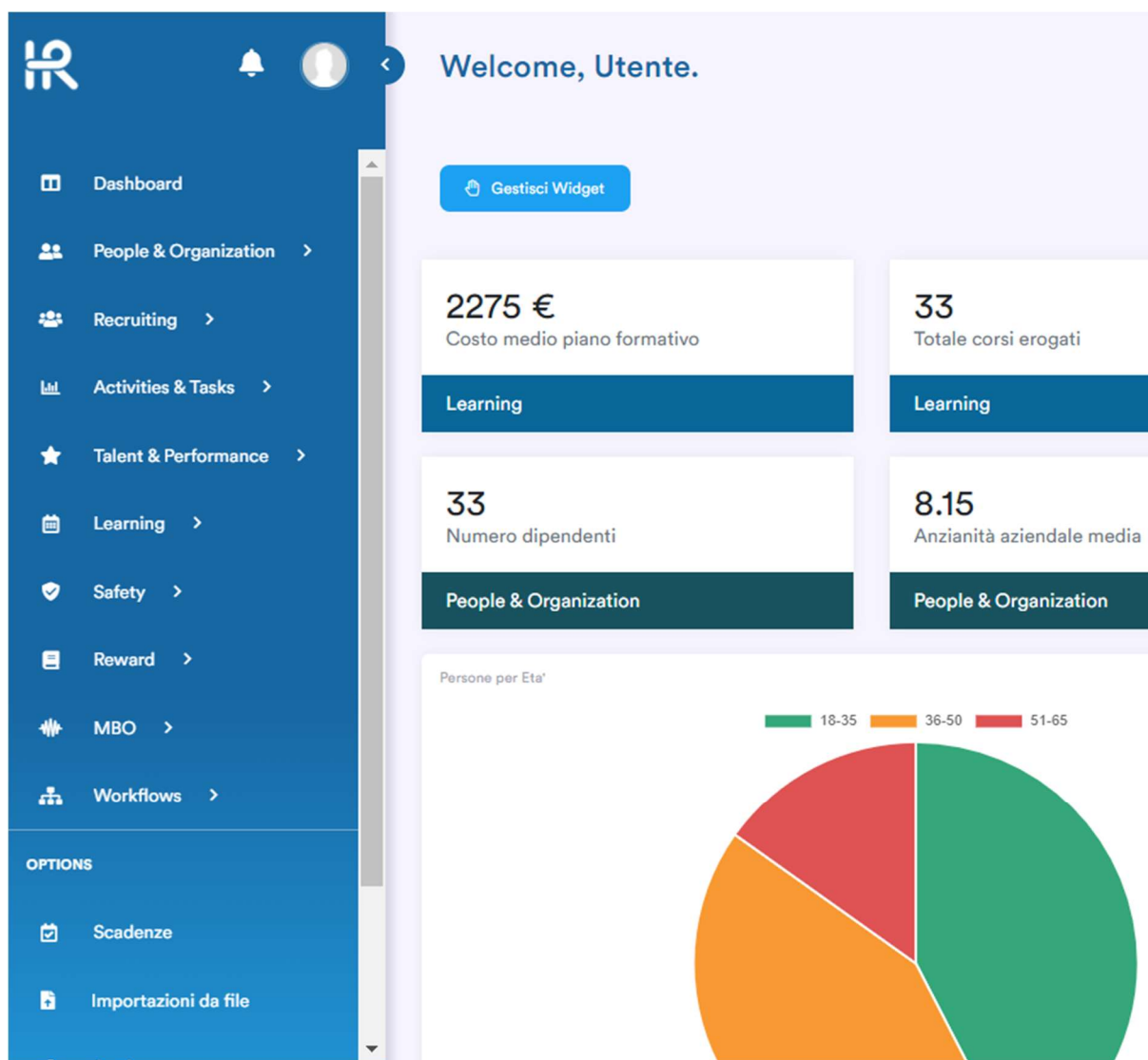


Figura 4.11- interfaccia hrtools.it

I principali moduli visibili nella sidebar sono quelli che possiamo vedere nella figura 4.8:

- Il modulo People & Organization prevede le funzionalità necessarie per registrare un nuovo utente.

- Recruiting prevede la candidatura per un ruolo.

- Activities and Tasks prevede di aggiungere le nuove attività e compiti per un mansione.

- Talent and performance prevede la parte di valutazione, conoscenze, abilità.

- Learning prevede conoscenze, grading conoscenze, piani

formativi, catalogo formativo.

- Saftey prevede le visite mediche, fattori di rischio per ogni ruolo.

- Rewards prevede le compense, dinamica retributiva, dati contrattuali e dati retributiva.

- Scadenze prevede di ricordare le scadenze di un corso formativo, di un certificato per un dispositivo di sicurezza e le scadenze per le visite mediche.

- Importazione di file che prevede di importare i file nel database tramite un file excel.

- Setting prevede di modificare alcuni parametri di ogni modulo.

Tutti i moduli verranno descritti in dettaglio nel prossimo capitolo.

## 4.5 - Organizzazione Strutturale

Laravel fornisce di default una struttura delle cartelle e dei file molto pulita e ordinata, infatti propone per ogni modulo una divisione secondo il pattern MVC come mostrato nella figura 4.12, che rappresentano le cartelle principali come `app/controller`, `resource/views` e `app/Models` e un file `web.php` per le rotte(routes) di pagine web, che viene utilizzato per aggiungere dei nuovi contenuti a quelli standard, come ad esempio i widget. Dentro la cartella `resources/lang` è possibile creare dei file per rendere la piattaforma multilingua e alla fine `resources/js` che gestisce tutte le librerie di vuejs per front-end.

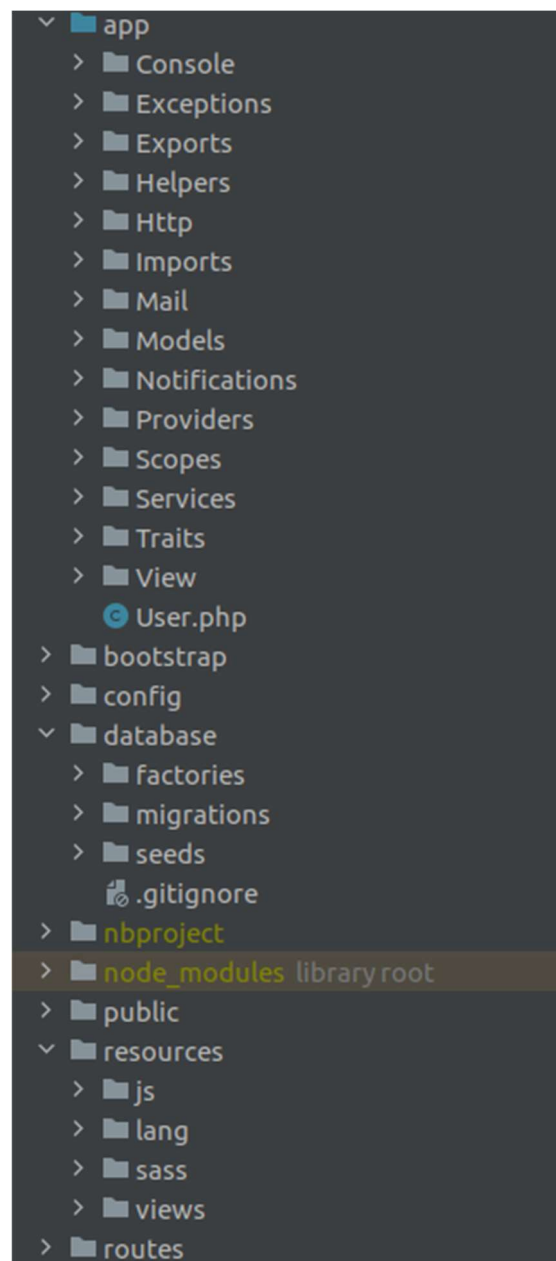


Figura 4.12 - Esempio di struttura delle cartelle

## Capitolo 5

### HRTTools(Human Resource Tools)

#### 5.1 - Introduzione

In questo capitolo verrà presentato HRTTools.it, la *web application* per la gestione delle risorse umane.

Nel prima parte si fornirà un quadro generale dell'applicativo e i moduli principali di hrtools in dettaglio con foto dell'interfaccia, e nel secondo paragrafo si fornirà un quadro generale sul brand name.

#### 5.2 - Human Resource Management

L'idea alla base di hrtools è quella di proporre un applicativo in modalità *software as a service* per la gestione delle risorse umane in ambito aziendale, che punti soprattutto sulla semplicità di utilizzo e sull'integrazione con le altre piattaforme di amministrazione del personale. In particolare, HRTTools.it è pensato per quelle aziende che hanno la loro strategia gestionale sulla misurazione e valutazione delle performance dei dipendenti in relazione al ruolo ricoperto, al fine di valorizzare chi ottiene i risultati migliori. L'URL attraverso cui i clienti accedono ad HRTTools.it è così composta: nome dell'azienda cliente +.hrtools.it. Ogni azienda dunque ha il suo accesso personalizzato all'applicativo, tramite una URL semplice.

L'applicativo offre agli HR manager gli strumenti necessari per la costruzione di un quadro esaustivo della situazione aziendale, che permetta di pianificare con cognizione di causa le strategie adeguate per puntare a un miglioramento costante del rendimento del personale. Nello specifico, una volta costruito il database con tutti i dati necessari, HRTools.it permette di impostare la valutazione in funzione delle competenze e conoscenze dei dipendenti, valutarne le prestazioni in base a particolari fattori legati alla posizione e impostare la politica salariale sulla base dei risultati raccolti.

### 5.3 - I moduli di HRTools.it

Le funzioni principali fornite da HRTools.it sono distribuite sui seguenti moduli: "People & Organization", "Activities and Tasks", "Talent and performance", "Recruiting", "Learning", "Safety", "Reward" e "Import".

#### 5.3.1 - Anagrafica (People & Organization)

Il modulo "Anagrafica" permette di creare un database con tutte le informazioni relative al personale dell'azienda, necessarie per il funzionamento degli altri moduli dell'applicativo. È suddiviso in tre sezioni, "Dipendenti" e "Historical". La Anagrafica raccoglie in una tabella i dati relativi ai dipendenti dell'azienda: cognome, nome, indirizzo email, data di nascita, data di assunzione, matricola, codice fiscale, posizione ricoperta, data a partire dalla quale ha cominciato a ricoprire tale posizione, responsabile di reparto, sede presso la quale il dipendente è impiegato.

Tutte le tabelle presenti in questo modulo possono essere compilate sia manualmente attraverso pagina web che attraverso l'importazione di file XLS o CSV, funzionalità che può rivelarsi molto utile per velocizzare il processo di popolamento del database. A loro volta, i dati contenuti nelle tabelle possono poi essere scaricati in formato XLS per crearne una copia di backup.

#### 5.3.2 - Activities and Tasks

Nel modulo "Activities and Tasks" vengono immagazzinati tutti i dati relativi alle mansioni e alle attività interne all'azienda.

Una volta compilato il database con l'elenco delle mansioni, delle attività e

dei compiti è possibile costruire una associazione tra una persona e il suo ruolo. Per ogni ruolo è possibile fornire un nome, una descrizione, dei fattori di rischio, un grado ecc.

### 5.3.3 - Talent and performance

Il modulo “Talent and performance” consente la creazione della banca di dati relativa alle skills, la sessione di valutazione, inoltre per ogni skill è possibile fornire un nome, descrizione, livello. E per ogni sessione di valutazione è possibile fornire un nome, una data inizio sessione, una data fine sessione e il tipo di questionario.

### 5.3.4 – Recruiting

Il modulo “Recruiting” consente la creazione di banca dati relativi alle campagne di reclutamento. Il database candidati permette di storicizzare e mantenere le informazioni di ogni persona entrata in contatto con l’azienda durante le campagne di selezione. Le informazioni sono consultabili in un profilo. Per ogni Recruiting è possibile fornire un nome, email, ruolo, telefono, c.f., titolo di studio e data di assunzione.

### 5.3.3 – Skills

Il modulo “skills” consente la creazione della banca dati relativa alle competenze necessarie per lavorare in azienda. Per ogni competenza è possibile fornire una descrizione, elencare i comportamenti attesi dai dipendenti per ognuna di esse e indicare il livello richiesto in funzione della posizione, in base al quale il dipendente verrà considerato idoneo al ruolo ricoperto. Anche in questo caso, i dati vengono raccolti in tabelle simili per forma e con la possibilità di importare il database da file CSV o esportarlo in formato XLS. Le informazioni di modulo “Skills” sono importanti perché sono utilizzate per il modulo “Schede di Valutazione”.

### 5.3.4 - Reward

Il modulo “reward” consente di pianificare la politica retributiva dell’azienda, sia per quanto concerne la parte fissa del monte salari che per quella variabile. Inoltre rende possibile la profilazione di percorsi di potenziale

crescita retributiva del personale in base all'equità e alla professionalità espresso. Una volta inseriti nel database i dati retributivi e contrattuali per ciascuno dei dipendenti, sarà possibile usufruire della sezione “Dinamica Retributiva”, che simula l'evoluzione a medio termine dello stipendio annuale. Anche in questo caso, i dati vengono raccolti in tabelle simili per forma e con la possibilità di importare il database da file CSV o esportarlo in formato XLS.

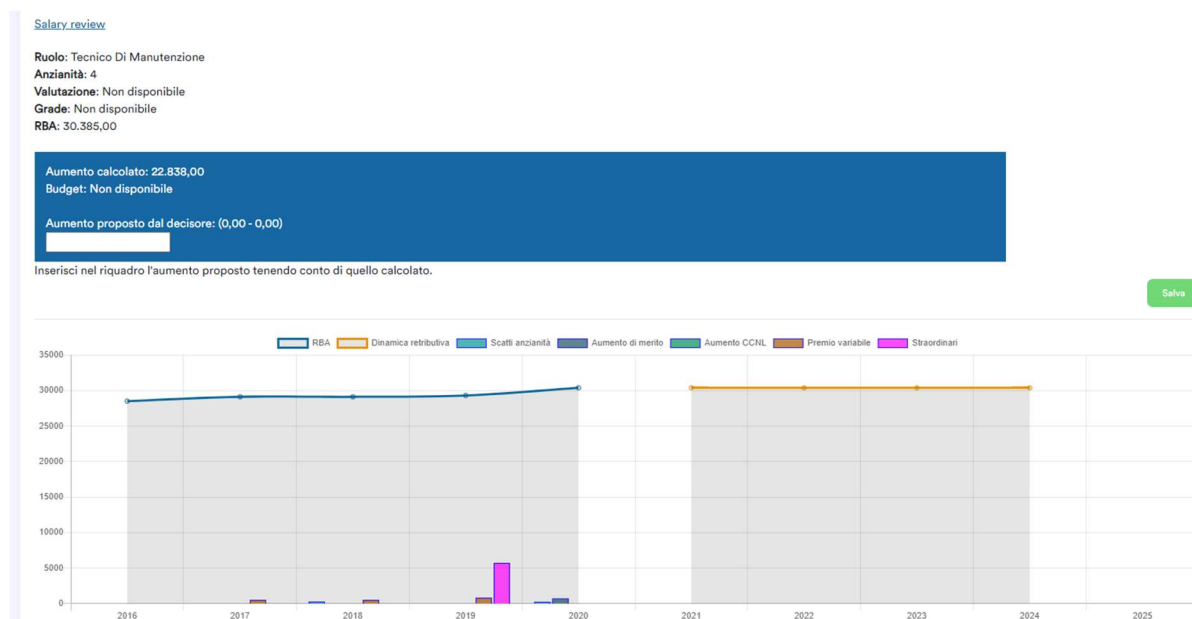


Figura 5.1 - Modulo *rewards*, Esempio di Dinamica retributiva

### 5.3.5 - Learning

Il modulo “Learning” consente la creazione di banca dati relativa ai piani di formazione, alla gestione dei piani formativi e dei relativi costi, delle lezioni e dei partecipanti. Condivisione della documentazione della formazione, per ogni piano di formazione è possibile fornire un nome, una descrizione, un codice corso, una data inizio corso, un luogo, le ore previste, un ente di formazione e un Docente. Anche in questo caso, i dati vengono raccolti in tabelle simili per forma e con la possibilità di importare il database da file CSV o esportarlo in formato XLS.

### 5.3.6 - Safety



Il modulo “Safety ” consente la creazione di banca dati relativa alla sicurezza dei dipendenti ed è formato dalla tabella visite mediche, dalla tabella dispositivi di sicurezza e fattori di rischio. Per ogni visita medica è possibile fornire un nome, una descrizione, una scadenza, e un esito. Ogni fattore di rischio è associato ad un ruolo e anche in questo caso, i dati vengono raccolti in tabelle simili per forma e con la possibilità di importare il database da file CSV o esportarlo in formato XLS.

## Conclusioni

Al termine della mia esperienza di tirocinio sono diventato consapevole di quello che tratta il mondo del lavoro e ho fatto uno dei lavori per la quale mi ero iscritto all'università. Grazie sia all'ambiente di lavoro sia ai colleghi, questo lavoro è stato davvero piacevole da imparare.

La creazione e la stesura del modulo People & organization sono state affrontate con funzioni che permettano un utilizzo semplice e intuitivo in modo da poter essere adoperate da chiunque si approcci a questo sistema.

Tramite la formazione che mi è stata data all'inizio del tirocinio da parte dei tutor, che mi hanno seguito durante tutto il mio percorso, l'implementazione di questo modulo non ha riscontrato grandi problemi anche grazie alla eccellente preparazione delle tabelle del database che hanno facilitato molto lo sviluppo e soddisfatto a pieno i requisiti proposti. La parte più complicata e necessaria per il completamento del progettazione è stata di creare una sistema di ricerca particolare, che ha richiesto una grande abilità in base di dati e vue js.

Gli sviluppi futuri del progetto riguardano innanzitutto la completa implementazione e messa in funzione dei nuovi moduli. Inoltre, HRTools vorrebbe realizzare in futuro un nuovo modulo del sito web che consentirà ai clienti di segnalare i problemi per rendere HRTools ancora piu user-friendly. Emme delta mi ha già proposto della possibilità di proseguire la nostra collaborazione, tale proposta risulta essere un'ottima opportunità per ampliare le mie conoscenze nel campo di sviluppo web.

## Bibliografia

- [1] “Hrtools”: <https://emmedelta.it/>
- [2] “Metodologia Agile”: [https://it.wikipedia.org/wiki/Metodologia\\_agile](https://it.wikipedia.org/wiki/Metodologia_agile)
- [3] “Applicazione Web”: [https://it.wikipedia.org/wiki/Applicazione\\_web](https://it.wikipedia.org/wiki/Applicazione_web)
- [4] “Modello E/R”: [https://it.wikipedia.org/wiki/Modello\\_E-R](https://it.wikipedia.org/wiki/Modello_E-R)