



Università degli Studi di Bari
Facoltà di Informatica
Corso di Laurea in Informatica e T.P.S.



Tesi di Laurea in Sistemi Cooperativi

Sviluppo di un Registro Sanitario Elettronico Personale utilizzando il Web Framework “Laravel”

Relatore:

Chiar.mo Prof. Giovanni Dimauro

Correlatore:

Dott. Francesco Girardi

Laureando:

Giuseppe Barbato

Anno Accademico 2016/2017

Sommario

- INTRODUZIONE
- IL FRAMEWORK “LARAVEL”
- LA VIRTUAL MACHINE “HOMESTEAD”
- FASE DI SVILUPPO
- CONCLUSIONI



Obiettivo

Sviluppare una web application che abbia come punto di partenza il patrimonio di conoscenze già generato dal sistema R.E.S.P. andando tuttavia a risolvere varie problematiche presenti.

Il nuovo sistema generato sarà basato su di un framework al fine di trarre vantaggio in termini di:

- Manutenibilità
- Affidabilità
- Estendibilità
- Robustezza
- Sicurezza



Sistemi Informativi Sanitari

- Electronic Health Record (EHR)
- Dossier Sanitario Elettronico (DSE)
- Fascicolo Sanitario Elettronico (FSE)
- Personal Health Record (PHR)



Fascicolo Sanitario Elettronico

Rappresenta l'insieme dei dati e documenti digitali di tipo sanitario e socio-sanitario generati da eventi clinici presenti e trascorsi, riguardanti l'assistito.

Agevola il personale medico nel tracciare in maniera persistente l'anamnesi del paziente.

Evita la dispersione di documenti in formato cartaceo.

Consente di tracciare le terapie farmacologiche che l'assistito segue o ha seguito in passato.



Personal Health Record

Cartella clinica dove i dati clinici e le informazioni legate alla storica clinica sono gestite dal paziente stesso.

Non orientato molto alla consultazione delle informazioni da parte dei medici.

Include la possibilità di inserire rilevazioni di parametri vitali, esiti di esami diagnostici o di laboratorio, allergie, vaccini e altro ancora.



Registro Elettronico Sanitario Personale

Il sistema R.E.S.P. è un sistema sanitario ibrido prevalentemente incentrato sulla figura del paziente.



Funzionalità Del R.E.S.P.

LATO PAZIENTE

- Gestione anagrafica e contatti
- Anamnesi
- Diagnosi
- Caricamento di files diagnostici
- Taccuino paziente
- Gestione delle visite effettuate
- Calcolatrice medica

LATO CARE PROVIDER

- Gestione dei propri pazienti (accesso ai dati in base al livello di restrizione assegnatogli)
- Gestione delle strutture sanitarie in cui esercita
- Modifica ed aggiornamento dei profili dei propri pazienti (se consentitogli)



Problematiche

L'assenza di una struttura rigida alla base ha causato problemi di diversa natura. Le principali categorie sono:

- Complicazioni in fase di sviluppo
- Livello generale di sicurezza
- Manutenibilità
- Difficoltà nell'effettuare testing



Framework

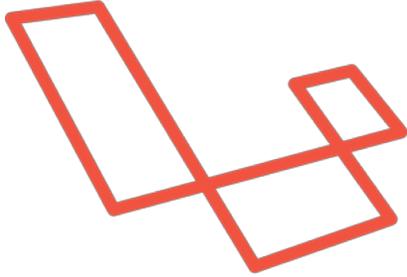
Al fine di risolvere le problematiche citate e per ottenere una web application di qualità si è deciso di costruire il sistema utilizzando un framework.

L'utilizzo di un framework comporta numerosi vantaggi rispetto allo sviluppo tradizionale di tutte le singole componenti di un sistema ed incentiva ad un approccio più strutturato.

Dopo un'attenta fase di selezione, si è deciso di utilizzare il web framework "Laravel"



Laravel



Web Framework, di tipo MVC, scritto in PHP

Creato nel 2011 da T. Otwell e A. Xhelo

Riportato da numerose fonti come il framework PHP più utilizzato al mondo

Vasta comunità di sviluppatori e disponibilità di componenti aggiuntivi

Aggiornato con alta frequenza



Laravel – Alcune Caratteristiche

- Integrazione di Eloquent ORM (Object-Relational Mapping) per l'interazione con il database
- Sistema di Routing
- Gestione delle views tramite il templating engine “Blade”
- Strumenti nativi per l'innalzamento dei livelli di sicurezza dell'intero sistema



Homestead

Utilizzato insieme a Git (per il versioning del sorgente) assicura che tutti gli sviluppatori possano lavorare sulla stessa versione del codice e sullo stesso ambiente di sviluppo.

È una Vagrant box basata sulla distribuzione Linux Ubuntu che incorpora al suo interno tutte le tecnologie necessarie per lo sviluppo con Laravel (PHP 7, MySQL, Nginx, Node, ecc...).

Idealmente, in fase di rilascio, si potrebbe trasferire direttamente Homestead su server e la web application sarebbe up senza necessità di ulteriori configurazioni.



Fase Di Sviluppo

Dopo la fase di progettazione si è passati alla fase implementativa vera e propria.

Poiché Laravel è basato su pattern MVC sono stati creati i vari models, views e controllers. All'interno di quest'ultimi, grazie a Eloquent ORM è stato possibile interagire con il database senza la necessità di svolgere delle query esplicite ma con un approccio più Object-Oriented.

```
$paziente = Pazienti::where('id_utente', Auth::id())->first();  
$paziente->paziente_nome;
```



Fase Di Sviluppo

Per quanto riguarda la gestione delle views, il Blade System ha permesso di gestire l'intero layout di base in poche righe di codice.

Inoltre l'utilizzo dei tag nativi del template engine ha permesso di evitare l'inserimento di codice PHP puro, riducendo drasticamente la verbosità del codice.

```
1 <!DOCTYPE html>
2 <html lang="{{ app()->getLocale() }}">
3   <head>
4     @include('includes.template_head')
5   </head>
6
7   <body>
8     <!-- Include il template dell'header -->
9     @include('includes.template_header')
10
11     <!-- Include il template del menu laterale sinistro -->
12     @include('includes.template_menu')
13
14     <!-- Carica il contenuto delle sezioni principali -->
15     @yield('content')
16
17     <!-- Include la sidebar destra solo se si è loggati come pazienti -->
18     @if(Auth::user()->getRole() == Auth::user()::PATIENT_ID)
19       @include('includes.template_sidebar')
20     @endif
21
22     <!-- Include il template del footer -->
23     @include('includes.template_footer')
24
25   </body>
26 </html>
```

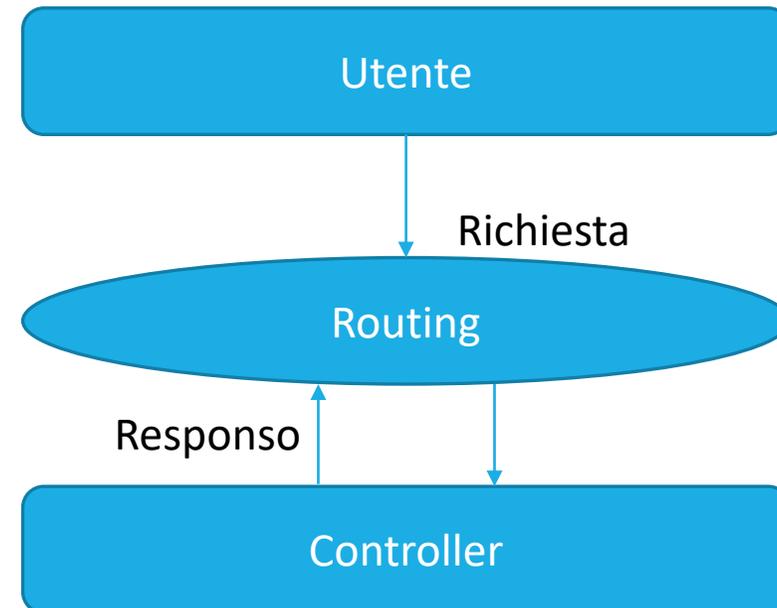


Fase Di Sviluppo

Altro aspetto curato è stato quello del routing per la gestione delle richieste partite dall'utente.

Ogni richiesta viene gestita a seconda del suo tipo (GET, POST, ecc...) e solo successivamente viene smistata ad un controller.

Tra i vantaggi offerti da questo meccanismo c'è la possibilità di raccogliere tipologie di richieste simili all'interno di middleware adattabili a seconda delle esigenze.



Conclusioni

La web application creata può senza dubbio essere considerata come un applicativo stabile, robusto e facilmente aggiornabile.

L'approccio utilizzato favorirà in futuro lo sviluppo da parte di nuovi developers ed ha innalzato drasticamente la manutenibilità e leggibilità dell'intero sistema.

Inoltre si è generato un sistema in grado di tutelare ad alto livello i dati immessi al suo interno; questione centrale all'interno di un sistema informativo sanitario come R.E.S.P.



Grazie per l'attenzione