Proxecto: Web Dietistas

Documentación

Autor: Diego Dosil Canosa

Grupo: DAW2 curso 2019-2020

I.E.S. Fernando Wirtz Suárez

Profesor responsable: Manuel Corbelle Mejuto

17/02/2020

Versión 00.01

Táboa de contido

1 Descripción do proxecto e definición de requisitos.

1.1 Prefacio: Obxectivos de aprendizaxe

Un dos principais obxectivos deste proxecto é o de pór en práctica e ampliar os coñecementos teórico-prácticos adquiridos no Ciclo Superior de Desenvolvemento de Aplicacións Web.

Por isto me decido á proposición deste proxecto, xa que me vai permitir traballar sobre coñecementos adquiridos en módulos do Ciclo como Linguaxes de Marcas, Deseño de interfaces web, Bases de Datos, Programación do lado do servidor ou Programación do lado do cliente

1.2 Obxectivos da aplicación

Deseñar e implementar un sitio web que permita a dietistas xestionar clientes. Isto permitirá que os dietistas poidan xestionalos tanto desde a súa clínica como desde o domicilio do cliente.

1.3 Requisitos da aplicación

Deben existir 3 roles claramente diferenciados: Administrador, Dietista e Cliente. Un dietista pode tratar a cantos clientes desexe, pero un cliente só o poderá ser tratado por un único dietista.

Web Dietistas debe ser un sitio web totalmente funcional, polo tanto terá que ser publicado na súa propia web, sendo así accesible aos dietistas desde os domicilios dos clientes de ser necesario.

Os clientes e os dietistas deberán poder comunicarse a través da web, consultar a súa evolución e xestionar as súas citas.

Os dietistas deben poder ter acceso a unha fonte de datos fiable de composición de alimentos.

Os dietistas deben poder engadir dietas á web e asociar os valores nutricionais totais destas receitas. Estas dietas poden ser compartidas polos dietistas cos seus clientes.

Débense ter en conta as intolerancias e alerxias alimentarias dos clientes. Estes deben informar ao seu dietista e este debe incluilas nos datos do cliente.

Débense ter en conta as enfermidades dos clientes. Estes deben informar ao seu dietista e este debe incluilas nos datos do cliente.

Poderase activar ou desactivar usuarios segundo sexa necesario.

2 Recursos a empregar

2.1 Aplicacións a empregar

Para o desenvolvemento deste proxecto será necesario empregar o seguinte software:

- Xampp→ Paquete que integra o servidor web Apache, intérprete para a linguaxe de programación PHP e o sistema de xestión de base de datos MySQL (MariaDB).
- No-IP DUC→ Servicio de DNS dinámico de No-IP.com. Será utilizado para poder ter un dominio asociado á nosa IP local aínda que esta sexa dinámica
- Sublime Text 3→Editor de código.
- ProjectLibre 1.8.0→Software de administración de proxectos
- MS Word 2007→ Editor de textos para elaborar a documentación do proxecto
- MS Visio→ Programa de debuxo vectorial para a creación dos diagramas necesarios para este proxecto
- Laravel→Framework para desarrollo de aplicaciones PHP
- Composer→Manejador de dependencias para PHP
- Git→Software de control de versións
- GitHub→Plataforma de desarrollo colaborativo.

O obxectivo é non ter que pagar polo software empregado e, dentro das opcións que hai, decanteime por Laravel e Git por ser os máis demandados polas empresas no ano 2019 en Galicia (fonte: Infojobs).

Empregaranse as librerías JavaScript: JQuery, Select2.js e Chart.js

2.2 Presuposto

- Software necesario.- Non será necesario ningún tipo de gasto. Intentarase o uso de aplicacións e librerías gratuitas e con licencia open-source.
- Hardware necesario.- Non será necesaria a inversión en novo hardware. Será necesario un PC para o desenrolo do proxecto e un móbil para as probas.
- Equipo humano.- Non será necesaria a incorporación de máis persoas para o desenvolvemento do proxecto. Será aconsellable a colaboración de xente para facer as probas de usuario.

3 Planificación

A selección de ideas para a realización do proxecto empeza o 30-9-2019 (data de inicio da Formación en Centros de Traballo) e o inicio deste proxecto é o 21-10-2019. A data de remate é o 17-2-2020 (data de finalización da FCT). A planificación temporal por tarefas pódese consultar no arquivo adxunto a esta documentación “Proxecto 2020.pod”.

4 Modelado conceptual

Pódese consultar o Diagrama Entidade-Relación no arquivo adxunto a esta documentación "Diagrama_ER.vsd". A continuación explícanse as cuestión menos obvias:

TÁBOA USUARIOS:

Ten unha relación reflexiva 1 a N. Un cliente pertence a un único dietista, pero un dietista pode ter máis de un cliente. Un dietista pertence a un único administrador, pero o administrador pode ter máis de un dietista.

O mail e o contrasinal son obrigatorios, xa que serán os que se pidan para iniciar sesión.

O campo "activado" indicará se o usuario está activado ou non.

O campo "dependencia" indicará o ID do usuario ó que pertence. Se é administrador, o valor será "0", xa que non pertence a ningún usuario. Se é dietista, o valor será "1", xa que o primeiro rexistro que debe existir é o do administrador. Se é cliente, o valor será o do ID do dietista ao que pertenza.

TÁBOA ALTERACIONES:

Indicará as posibles enfermedades, alerxias e intolerancias que un cliente pode sufrir.

TÁBOA USUARIOS-ALTERACIONES:

Xurde da relación N:M de Usuarios e Alteraciones

TÁBOA DIETAS:

Debe incorporar o ID do dietista que a elaborou. O sumatorio de cada dato da dieta debe incluirse no seu respectivo campo. A dieta pode ter dous estados: 0=non rematada, 1=rematada

TÁBOA ALIMENTOS:

Cada rexistro identificará a cada alimento que compón unha dieta e a súa cantidade.

TÁBOA CITAS:

Debe gardar en cada rexistro de cita os principais datos que permitan levar un seguimento da evolución do cliente (peso, IMC, etc.).

O campo "estado" poderá ter 3 valores: "pendente", serán as citas que están concertadas pero aínda non se realizaron. "realizada", serán as citas que se fixeron pero nas que aínda non se cubriron os datos recollidos nela. "completa", serán as citas que se fixeron e os datos da visita xa están cubertos.

5 Requisitos funcionáis

Tal e como se especificou nos requisitos, deben existir 3 roles. A continuación detállanse que funcionalidades terá cada un deles:

Administrador→Será un único usuario. Poderá haber máis se se desexa, pero para rexistralos, farase directamente na base de datos. Terá o valor "0" no campo "dependencia" e será o primeiro rexistro da táboa "usuarios". Poderá:

a) Dar de alta dietistas
b) Activar ou desactivar dietistas
c) Eliminar dietistas

Dietista→Será un usuario rexistrado como tal polo administrador. Terá o valor "1" no campo "dependencia", xa que vai depender sempre do administrador. Poderá:

a) Dar de alta clientes
b) Activar ou desactivar clientes
c) Eliminar clientes
d) Consultar, modificar e engadir os datos de evolución de peso e masa corporal de clientes
e) Crear dietas
f) Engadir alimentos ás dietas
g) Cambiar o estado das dietas (finalizada ou non finalizada)
h) Consultar, concertar, modificar ou anular directamente citas cos clientes
i) Contactar cos seus clientes por teléfono ou WhatsApp
j) Consultar a ubicación do domicilio do cliente online

Cliente→Será un usuario rexistrado como tal por un dietista. No campo "dependencia" terá o valor do id do dietista ao que pertence. Poderá:

a) Contactar co seu dietista por teléfono ou WhatsApp
b) Consultar as citas co seu dietista
c) Consultar a súa evolución de peso e masa corporal
d) Consultar as receitas enviadas polo dietista

6 Configuración do entorno de desenvolvemento e instalación do software necesario

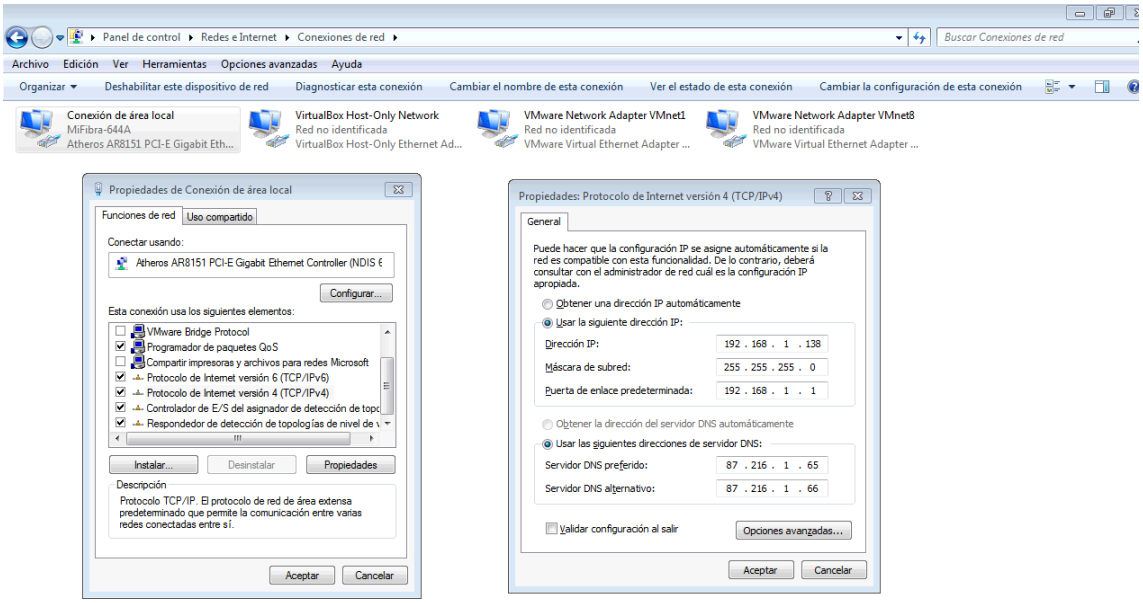
Antes de comezar, faise unha copia de seguridade de toda a carpeta "xampp".

Empezaremos instalando xampp desde https://www.apachefriends.org/es/index.html

Instalaremos a última versión, que a data da realización desta documentación é a 7.4.1 (PHP 7.4.1). Hai que ter en conta que empregaremos outro software que pode precisar esta última versión de PHP, coma Laravel.

Xa que imos crear un servidor local, imos ter facer as seguintes configuracións:

DNS sistema operativo:



DNS Router:

Router como web server:

Tendo en conta que a nosa IP é dinámica, para publicar en Internet o noso servidor local empregamos No-IP https://www.noip.com/

Para resolver a nosa IP, debemos descargar e instalar DUC https://www.noip.com/download?page=win

Configuración de Xampp:

Httpd.conf:

ServerName localhost:80

<Directory />

AllowOverride none

Require all Granted

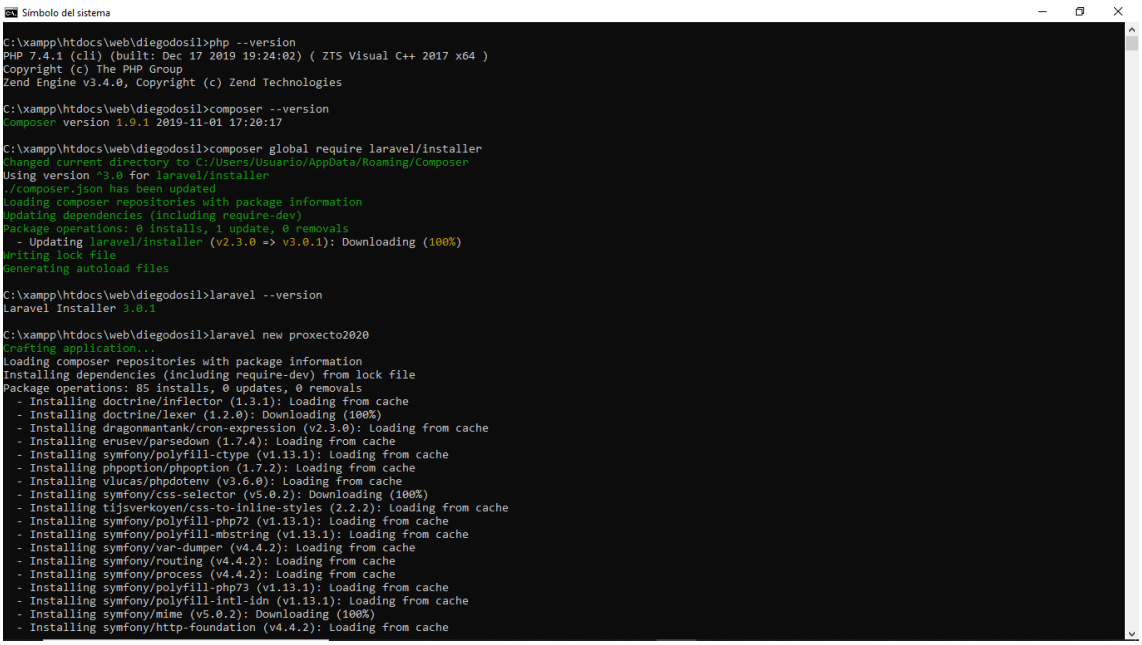
</Directory>

DocumentRoot "C:/xampp/htdocs/web/"

<Directory "C:/xampp/htdocs/web/">

Unha vez configurado o servidor, instalamos composer no directorio “diegodosil” desde https://getcomposer.org/download/

Comprobamos que temos todo o necesario para continuar, instalamos Laravel e creamos o noso novo proxecto como “dietistas”:



Agora intentamos crear o sistema de autenticación con php artisan make:auth, pero o comando quedou obsoleto a partir da versión 6 de Laravel, así que seguimos os seguintes pasos:

Con Composer instalamos o paquete ui de Laravel:

```
composer require laravel/ui
```

Agora creamos o sistema de autenticación:

```
php artisan ui vue --auth
```

npm install

npm run dev

7 Principais rutas

.env→Configuración da base de datos

App→Modelos

App/Http/Controllers→Controladores

App/Http/Middlewares→Middlewares

Database/migrations→Migracións para a bases de datos

Database/seeds→Población da base de datos

Docs→Documentación do proxecto

Public/css→Arquivos CSS a empregar

Public/js→Arquivos JavaScript a empregar

Resources/views→Vistas

Resources/views/components→Compoñentes Blade para as vistas

Resources/views/plantillas→Plantillas Blade das vistas

Resources/views/ layoutxeral→Layout común a tódolos roles

Resources/views/ layoutadmin→Layout para o rol administrador

Resources/views/ layoutcliente→Layout para o rol cliente

Resources/views/ layoutdietista→Layout para o rol dietista

Resources/views/admin→Páxinas do administrador

Resources/views/cliente→Páxinas dos clientes

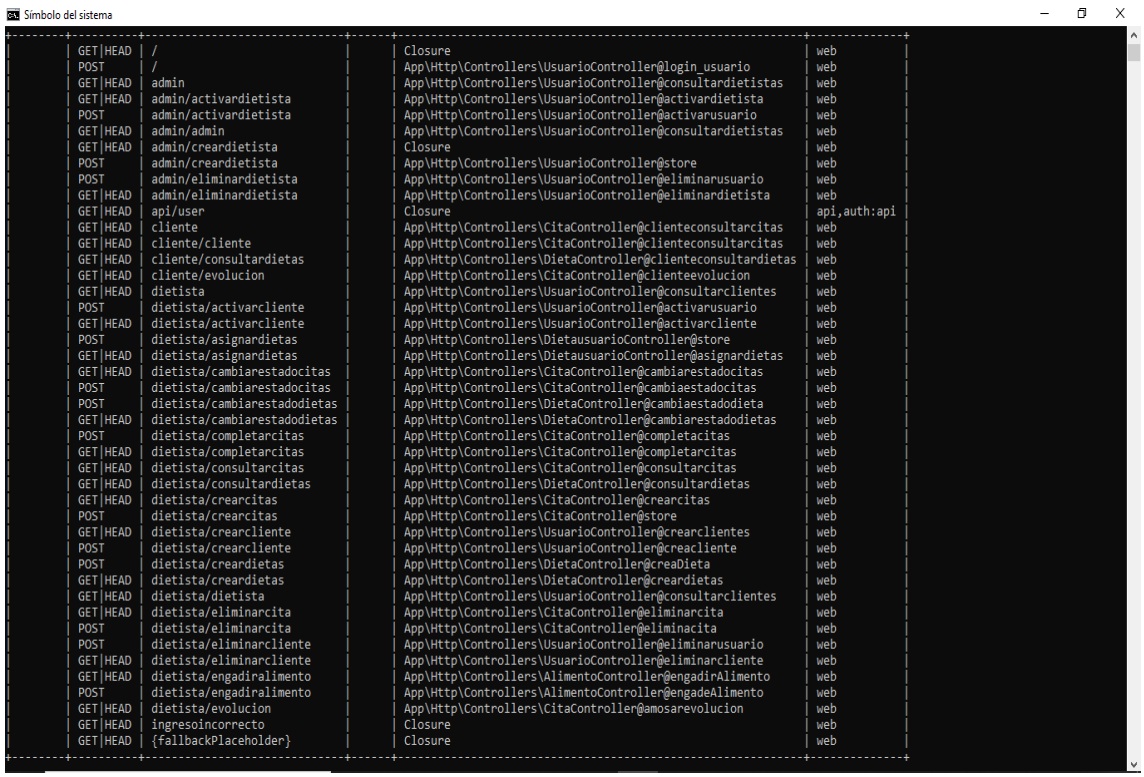
Resources/views/dietista→Páxinas dos dietistas

Resources/lang→Traduccións

Routes/web.php→Asignación de rutas

Storage/logs→Arquivo de logs de Laravel

As rutas das vistas e distintas chamadas a controladores pódense ver na seguinte imaxe:



8 Base de datos

Creamos unha base de datos baleira en MySQL á que chamaremos “dietistas”.

Artisan provéenos dunha serie de comandos que nos van facilitar enormemente o traballo. Podemos crear os modelos das nosas táboas do seguinte xeito:

php artisan make:model NomeDaTáboa

As migracións son un tipo de control de versións, pero para base de datos. Vannos permitir modificar o esquema da base de datos, poblar con datos as táboas, facer rollback, etc.

php artisan make:migration NomeDaTáboa

Se queremos que se faga a migración á base de datos:

php artisan migrate

Artisan tamén nos permite crear o esqueleto básico (CRUD) para os controladores:

php artisan make:controller NomeDaTaboa –resource

php artisan make:model Usuario

php artisan make:controller UsuarioController --resource

php artisan make:migration usuarios_migration --create=usuarios

php artisan migrate

9 Detalles da implementación

Ao non implementar o sistema propio de autenticación de Laravel, o servidor debe facer a comprobación en tódalas páxinas mediante sesións.

Tendo en conta que o servidor é o propio ordenador, emprégase o framework AMP como acerelador de páxinas.

Para a conexión coa base de datos de Bedca, emprégase a librería cURL. Examinando as chamadas que fai na súa propia web poiden levar a cabo as seguintes peticións de servicio necesarias para o correcto funcionamento do sistema:

- Obtención de tódolos alimentos
- Obtención dun elemento nutricional concreto dun alimento concreto

Para este proxecto requírense unha serie de elementos nutricionais, pero non todos os que Bedca aporta. Prevendo que en futuras versións se poidan necesitar outros elementos, necesitarase modificar o array especificado en AlimentoController.php, engadindo o identificador de Bedca para o elemento.

Durante a fase de probas póidose comprobar que en distintos equipos e navegadores a carga asíncrona das librerías JavaScript Chart.js e Select2.js, provocaba erros. Por isto, a carga destas librerías non é asícrona. En futuras versións habería que evaluar a posibilidade de servir as librerías desde o propio servidor da aplicación.

10 Probas

Realizáronse probas nos seguintes navegadores:

- Mozilla Firefox v73.0
- Microsoft Edge
- Opera
- Google Chrome v80

Tamén se fixeron probas en Smartphones con S.O. Android e navegadores Chrome e Firefox

11 Conclusións e liñas de traballo en futuras versións

11.1 Conclusións

O traballo realizado con este proxecto permitiume, por un lado, refrescar coñecementos xa adquiridos no ciclo e, por outro, ter unha ardua aprendizaxe en temas que nunca tocara coma os seguintes:

- Emprego de Laravel como framework de desenvolvemento. Tras unha ardua aprendizaxe, quedoume claro que é un framework moi potente e que facilita en gran medida o desenvolvemento e rutilización de código.

- Emprego de librerías JavaScript, coma Chart, Select2, Alertify e Momentjs, que permiten unha mellor experiencia de usuario.

- Enlazar ao servicio web de Bedca.

- Crear un servidor web doméstico.

Todo este traballo resultou ser moi satisfactorio e non pode ser máis positivo. Permitiume pór en práctica, como dixen antes, os coñecementos do ciclo, pero, además incluir novos coñecementos adquiridos na FCT e outros que foron necesarios mediante documentación e estudio.

O código desenvolvido, sen dúbida, hame ser de utilidade no meu futuro laboral, xa que nel hai solucións a problemas moi concretos e implementación de cuestións que soen empregarse.

11.2 Liñas de traballo en futuras versión

- Aplicación multiidioma
- Modificación do sistema de dietas, poidendo facerse semanal e separando os alimentos por comida (almorzo, xantar, etc)
- Obtención dun certificado SSL
- Conexión de dietista e cliente mediante Web-RTC

12 Anexos

12.1 Control de versións

As copias de seguridade son subidas a GitHub cunha descripción que permite saber cal é o principal cambio realizado. Por orde de creación son as seguintes:

- Primeira copia de seguridade→Creación do proxecto e estructura básica de directorios
- Inclusión da documentación→Inclúese toda a documentación elaborada
- Migrations e modelos→ Creación das migrations para crear as táboas da base de datos e os modelos desas táboas.
- Controladores→Creación dos controladores básicos. Creación da vista principal. Creación de plantillas e compoñentes.
- Vistas administrador→Creación de tódalas vistas do administrador. Modificación da base de datos, engadindo nas migrations os campos “created_at” e “updated_at”. Modificación do controlador de usuarios.
- Inicio vistas dietista→Creación das primeiras vistas do dietista. Modificación da táboa “citas” engadindo nas migrations o campo “estado”. Modificación do controlador de citas.
- Continuación vistas dietista→Creación de vistas de citas de dietista. Vista de evolución de clientes para dietistas. Modificación do menú de dietistas con novas opcións. Adaptación de deseño a móbil.
- V1→Creación do resto de vista de dietistas. Creación das vistas de cliente. Correccións de erros.