

MAYO 2021

PLATAFORMA DE EMPLEO DESARROLLADORES ARTTI

APP WEB

Elaborado por
Juliana Fajardo

Introducción

En la actualidad, en la sociedad la información y de la comunicación juega un papel muy importante en el desarrollo de esta.

Diariamente la información digital, es manejada por millones de personas sin importar su rango de edad, raza o religión.

Los usuarios necesitan enfrentarse a la reducción de problemas, no solo como usuarios sino también proyectarse hacia un futuro, donde la creatividad y la innovación sean el enlace hacia su vida cotidiana, por eso la necesidad de crear instrumentos que ayuden a superar estos inconvenientes, teniendo en cuenta la importancia de las nuevas tecnologías en el mundo globalizado.

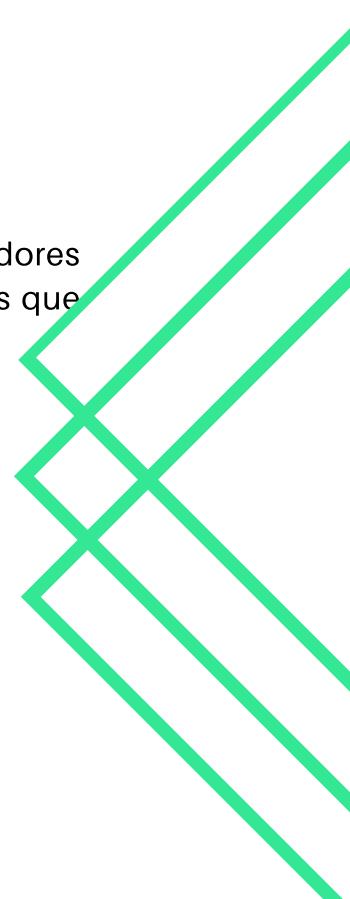
Por esta razón hemos detectado la problemática que presentan muchas empresas al no poseer una página web y la utilidad que esta brindaría.

Y la problemática para encontrar talento preparado para desarrollar sus soluciones tecnológicas.

por esta razón hemos creado una solución APP web para poder poner en contacto a desarrolladores y empresas o personas naturales que necesiten de estos servicios de desarrollo tecnológico.

Objetivo

Crear una plataforma que permita ofrecer un espacio para que reclutadores de empresas puedan postular sus ofertas laborales y los desarrolladores que estén buscando trabajo puedan postularse una solución a un click.



Alcance

XAMPP

Es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows/Linux, como sistema operativo;
- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un XAMPP permite servir páginas HTML a internet, además de poder gestionar datos en ellas, al mismo tiempo un XAMPP, proporciona lenguajes de programación para desarrollar aplicaciones web.

LAMP es el sistema análogo que corre bajo ambiente Linux WAMP Server es un completísimo paquete de software que sirve para instalar y configurar fácilmente un sistema el popular servidor Web Apache, juntamente con el lenguaje de programación PHP y el servidor de base de datos MySQL. Todo lo necesario para instalar un servicio de base de datos web, reunido en una única aplicación.

LARAVEL es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5, PHP 7 y PHP 8. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

En el caso de (nombre de la plataforma) es MVC (modelo vista controlador)

Alcance

Laravel Jetstream

Es la plataforma perfecta de scaffolding que se encuentra basado en framework PHP. Esto le permite ofrecer una gran cantidad de opciones para crear aplicaciones con Laravel. Cuenta con muchas funciones, y entre las más importantes tenemos:

- Soporte de API
- Autenticación de dos factores
- Posibilidad de verificar mediante correo electrónico
- Gestionar varios equipos

Las ventajas poseen trabajar con Laravel Jetstream

Base sólida

Uno de los objetivos de un programador es ahorrar tiempo, teniendo una base sólida para comenzar nuevos proyectos es más que suficiente. Algunos sistemas tan simples como el gestor de sesiones y doble factor de sesión llevarían horas de programación, inclusive para los que no son tan experimentados puede que le tome días. Pero con Laravel JetsStream sus módulos se lo permitirán realizar de manera sencilla y eficiente.

Migration Squashing

En español estas palabras quieren decir “Aplastamiento de migración” lo que quiere decir que mientras la app se haga más grande, es decir, la trabajemos mucho, la misma acumula migraciones con el pasar del tiempo. Pero la buena noticia es que a pesar de que el directorio de migraciones se llene de cientos de archivos, puedes utilizar MySQL o PostgreSQL para usar Migration Squashing y hacer que todo esto se convierta en un solo archivo SQL. Con tan solo la ejecución del comando de esta ventaja, Laravel ejecutará la tarea.

Función de mantenimiento

Esta función permite que el desarrollo de aplicaciones web que no esté terminado, pueda mostrarse en dicho estado a los clientes, de manera que puedan ver que el servicio se encuentra offline por mantenimiento. Laravel cuenta con un tutorial de como poder utilizar la versión 5 del programa para activar el modo mantenimiento.

DISEÑO WEB.

Creamos contenido atractivo que todos los usuarios en línea sabrán apreciar.

Es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web. El diseño de páginas web trata básicamente de realizar un documento con información hiperenlazado con otros documentos y asignarle una presentación para dispositivos de salida. La nueva tendencia orienta a que las páginas Web no sean solo atractivas para los internautas. También que sea popular entre ellos, una de las formas más viables para esto es realizar un diseño interactivo y moderno a la vanguardia y tratando de cubrir las necesidades de nuestros visitantes.

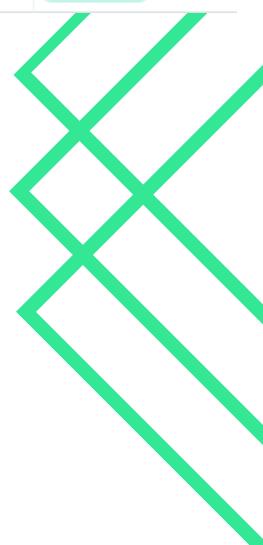
Fases del Proyecto

Consta de cuatro fases: Planeación, Diseño y ejecución, mantenimiento y evaluación del proyecto.

Planeación

Cronograma de actividades

| PLATAFORMA DESARROLLADORES | | | | |
|--------------------------------|--|---------------------------------|----------------------|-----------|
| PROYECTO | COMPETENCIAS | EQUIPO | Add or import | |
| A TAREA | DESCRIPCION | ESPECIFICACION | FECHA DE TERMINACION | Juliana |
| Definición arquitectura | stack LAMP - Proyecto Plataforma de empleo para desarrolladores | | 3/5/2021 | TERMINADO |
| Crear GIT | | | 3/5/2021 | TERMINADO |
| Crear Entorno de trabajo | | | 3/5/2021 | TERMINADO |
| Mock-up (plantilla) | •Página Principal • Inicio de sesión • Registro de un usuario • Edición de perfil • Vista de perfil d... | | 4/5/2021 | PROCESO |
| Diseño base de datos | Deberás diseñar e implementar una base de datos que permita desarrollar la solución descrita a c... | | 4/5/2021 | TERMINADO |
| Vista pagina principal | Deberá contener un menú que permita navegar a través de las categorías, así como botones de a... | | 5/5/2021 | TERMINADO |
| Vista inicio de sección | Deberá contener un formulario que pida correo y contraseña para iniciar la sesión, en caso que el ... | | 6/5/2021 | TERMINADO |
| RECLUTADOR Vista Registr... | Crear una visual donde un usuario pueda escoger el tipo de registro que quiere hacer en la aplicaci... | Como capa de seguridad l... | 7/5/2021 | TERMINADO |
| DESARROLLADOR Vista Re... | Crear una visual donde un usuario pueda escoger el tipo de registro que quiere hacer en la aplicaci... | Como capa de seguridad l... | 10/5/2021 | TERMINADO |
| RECLUTADOR Vista Edición... | Deberán crear un formulario que permita al usuario editar la información de su cuenta, y pueda c... | | 11/5/2021 | TERMINADO |
| DESARROLLADOR Vista Edi... | Deberán crear un formulario que permita al usuario editar la información de su cuenta, y pueda c... | | 11/5/2021 | TERMINADO |
| RECLUTADOR Perfil | Mostrar dinámicamente el contenido de la empresa y las vacantes disponibles de la misma: ○ Fot... | | 12/5/2021 | TERMINADO |
| DESARROLLADOR Perfil | Mostrar dinámicamente el contenido del perfil del desarrollador: ○ Foto de perfil ○ Años de expe... | Cargar CV y Descargar CV | 12/5/2021 | TERMINADO |
| Vista Categoría vacantes | Deberás crear unas categorías Deberán mostrar la información de las vacantes asociadas a la cate... | | 13/5/2021 | TERMINADO |
| Vista Detalle de vacante | Deberán mostrar el detalle de la vacante ○ Título de la vacante ○ Años de experiencia que se req... | Si un desarrollador ya inici... | 14/5/2021 | TERMINADO |
| Vista Panel de control de v... | Mostrar la información de las vacantes registradas del reclutador: ○ Título de la vacante ○ Estado... | | 17/5/2021 | TERMINADO |
| Vista Registro de una vaca... | Crear un formulario que registre una vacante con los siguientes campos: ○ Título de la vacante ○ ... | | 18/5/2021 | TERMINADO |
| Vista Candidatos | Crear una vista donde se muestren los candidatos que aplicaron a una oferta. plus (opcional): agre... | | 19/5/2021 | TERMINADO |
| Documentación | | | 20/5/2021 | TERMINADO |
| Informe | El informe debe contener: • Entre 10 y 20 páginas • Una introducción. • Una primera parte sobre:... | | 21/5/2021 | TERMINADO |
| Subir a producción y testiar | Para hacer pruebas y subir a producción | | 24/5/2021 | TERMINADO |
| Fecha de entrega | | | 25/5/2021 | TERMINADO |



Diseño

Mockup web y móvil

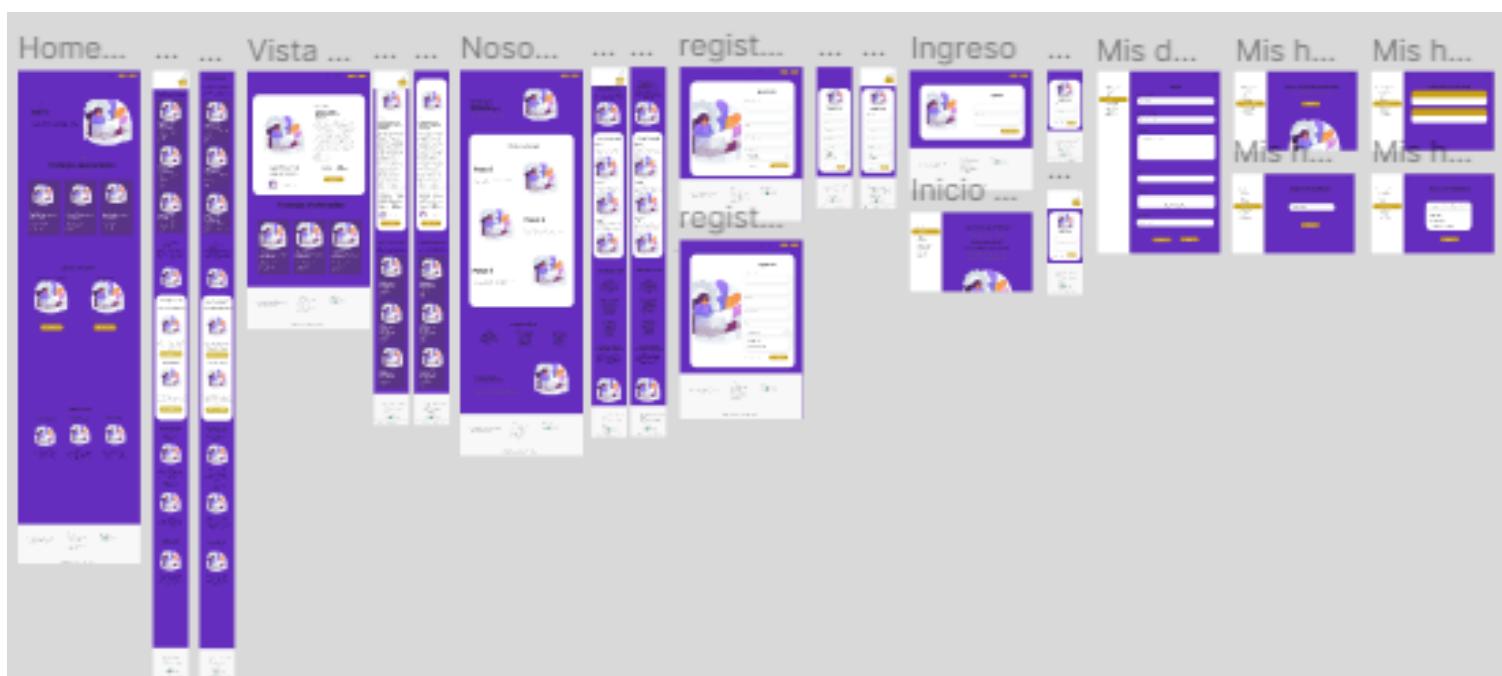
Autorización

Elementos de seguridad implementados

Estructuración del sistema de archivos

Funcionalidades clave específicas del proyecto

Mockup web y móvil



Mockup web y móvil

The mockup displays a website and a mobile application interface for ARTTI. The website header includes navigation links for Inicio, Nosotros, Ingresar, and Registrarse, along with a dark mode toggle icon. The mobile app interface follows a similar layout with a dark mode toggle icon in the top right corner.

Website Content:

- Sobre nosotros:** A section describing ARTTI as a Colombian company connecting software developers with leading sector companies.
- ARTTI:** The company's logo, featuring a stylized illustration of a person working at a computer with gears and leaves.
- Últimas vacantes:** A section listing job openings.
- Trabajos destacados:** A section titled "Trabajos destacados" with three cards, each showing a thumbnail of the ARTTI logo, the job title "DESARROLLADOR FULLSTACK NET JS BILINGÜE", and details like "Últimas vacantes" and "UBICACIÓN: COLOMBIA".
- Page Navigation:** Includes "Page 1 / 22", a search icon, and zoom controls (+/-).

Mobile Application Content:

- Últimas vacantes:** A section listing job openings.
- DESARROLLADOR FULLSTACK NET JS BILINGÜE:** A card detailing the job title, location "COLOMBIA", salary "200 COP", and currency "Moneda: COP".
- Page Navigation:** Includes "Page 1 / 22", a search icon, and zoom controls (+/-).

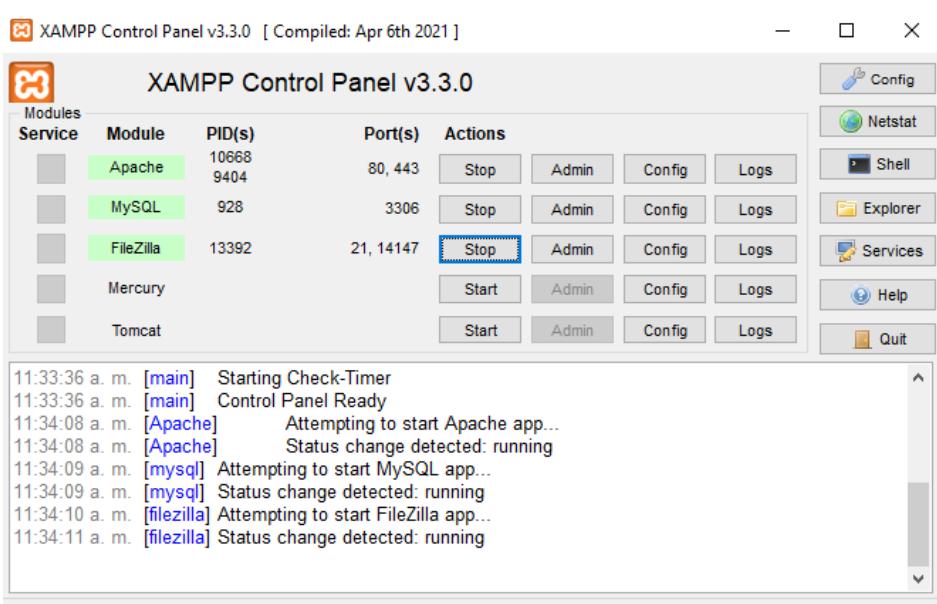
Autorización

Elementos de seguridad implementados

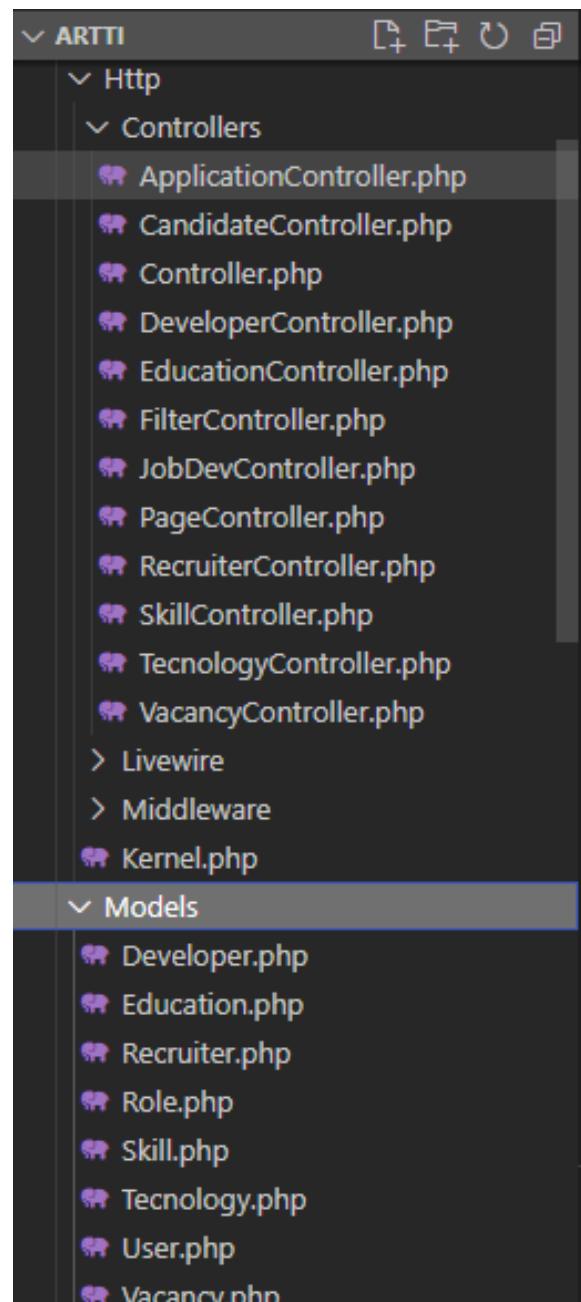
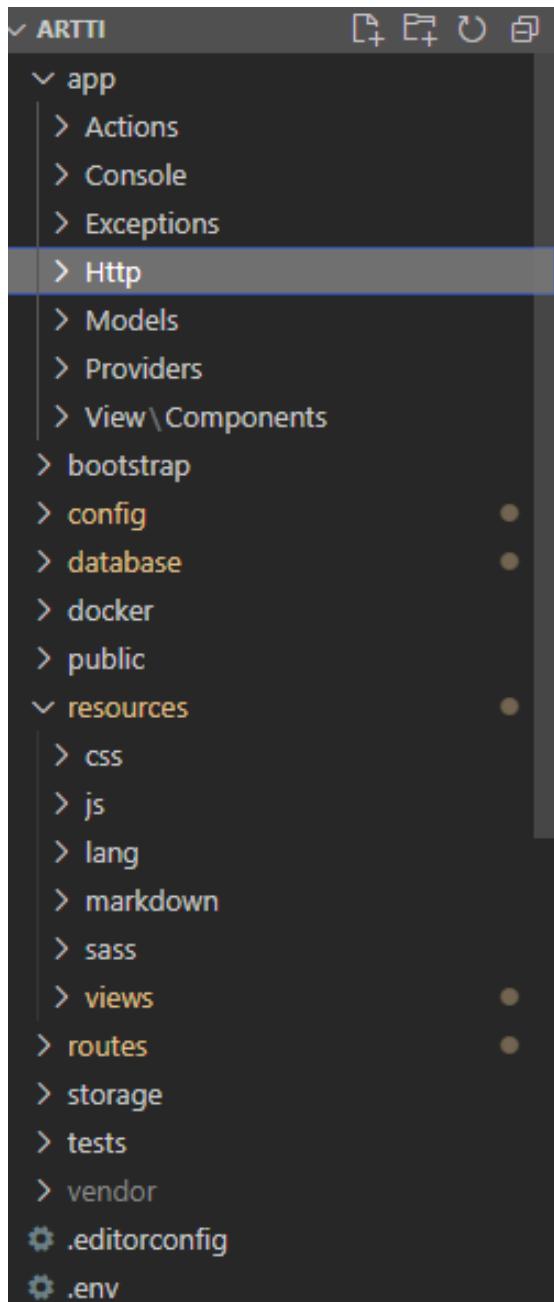


Laravel Jetstream

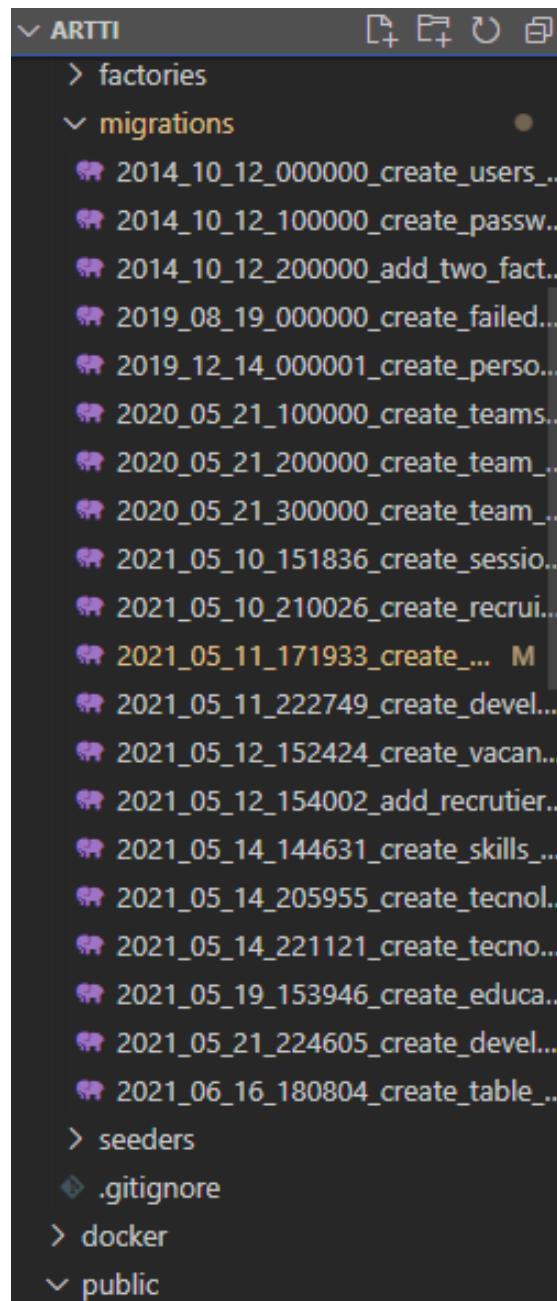
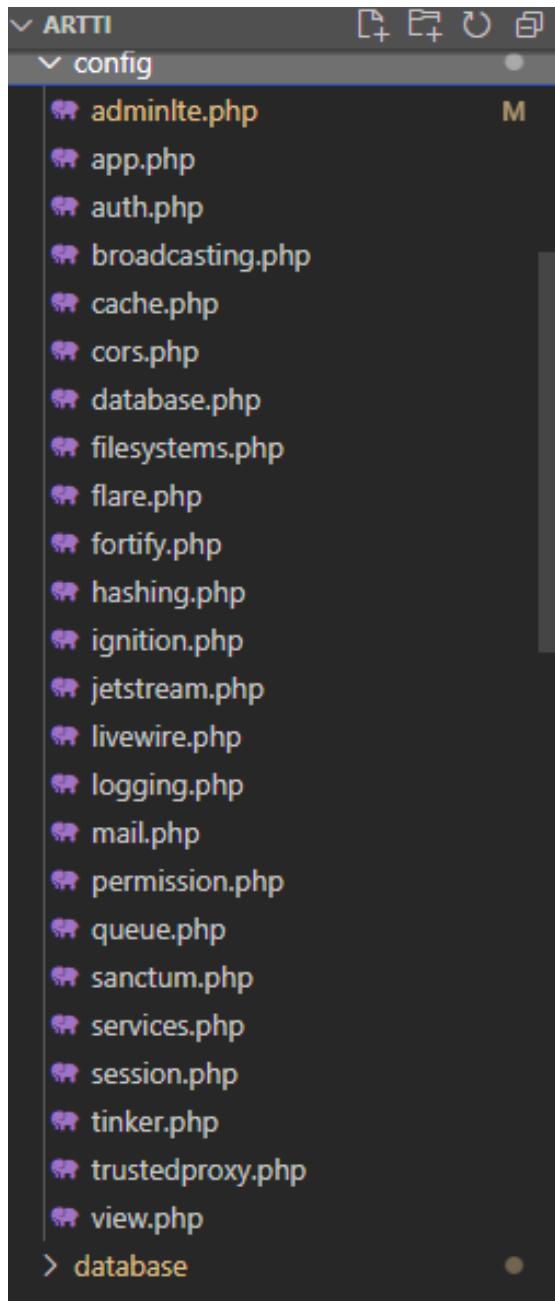
Tailwind CSS



Estructuración del sistema de archivos



Estructuración del sistema de archivos



Estructuración del sistema de archivos

```
ARTTI
views
  api
    api-token-manager.blade.php
    index.blade.php
  auth
    confirm-password.blade.php
    forgot-password.blade.php
    login.blade.php
    register.blade.php
    reset-password.blade.php
    two-factor-challenge.blade.php
    verify-email.blade.php
  developer
  errors
  jobDev
  layouts
  livewire
  profile
  Recruiter
  Vacancy
  vendor
  board.blade.php
  companydate.blade.php
  dashboard.blade.php
  navigation-menu.blade.php
  our.blade.php
```

```
ARTTI
> views
> routes
  api.php
  channels.php
  console.php
  jetstream.php
  web.php
> storage
> tests
> vendor
  .editorconfig
  .env
  .gitattributes
  .gitignore
  .styleci.yml
  artisan
  composer.json
  composer.lock
  package-lock.json
  package.json
  phpunit.xml
  prueba.php
  README.md
  server.php
  tailwind.config.js
  vendor.zip
```

Funcionalidades clave específicas del proyecto

Desarrollo e implementación Crear una solución que permita cumplir los siguientes requisitos:

- Página principal: deberá contener un menú que permita navegar a través de las categorías, así como botones de acceso para iniciar sesión o registrarse en la plataforma. Deberá tener una visual de las últimas vacantes agregadas y tener un buscador con filtros, donde pueda seleccionar, categoría, rango salarial, años de experiencia solicitados y ubicación. También deberán implementar un buscador que dé como resultado las coincidencias de las vacantes buscadas.
- Página de Inicio de sesión: deberá contener un formulario que pida correo y contraseña para iniciar la sesión, en caso que el usuario no ingrese la información correcta deberán mostrar una notificación indicando el motivo por el cual no puede ingresar. Deberán indicar en el mismo formulario un link que permita recuperar o cambiar la contraseña
- Página de registro: Crear una visual donde un usuario pueda escoger el tipo de registro que quiere hacer en la aplicación: Cómo Desarrollador: crear un formulario que pida al usuario:
 - Nombre Completo
 - Correo electrónico
 - Contraseña: mínimo 8 caracteres
 - Ubicación: País
- Aceptación de términos y condiciones Cómo Reclutador: crear un formulario que pida al usuario:
 - Nombre Completo
 - Correo electrónico
 - Contraseña: mínimo 8 caracteres
 - Empresa a la que pertenece
 - Ubicación: País

Funcionalidades clave específicas del proyecto

- Aceptación de términos y condiciones Como capa de seguridad los usuarios registrados recibirán un correo de confirmación de cuenta, si el usuario no confirma su correo no podrá interactuar en la plataforma, es decir como role de desarrollador no podrá postularse a las vacantes disponibles y como reclutador no podrán crear vacantes.
- Recuperación de contraseña: Deberán crear un formulario que permita recuperar o cambiar la contraseña. En cualquiera de los casos el usuario recibirá una notificación por correo electrónico para realizar cualquiera de las 2 acciones.
- Página de edición perfil de usuario (Desarrollador) : deberán crear un formulario que permita al usuario editar la información de su cuenta, y pueda completar su perfil agregando la siguiente información:
 - Foto de perfil (adjunto jpg o png, Máximo 200KB)
 - Años de experiencia
 - Estudios realizados
 - Habilidades
 - Tecnologías y/o lenguajes de programación
 - Portafolio
 - Proyectos destacados
 - Sobre mi
 - Certificaciones
 - Idiomas
 - CV (adjunto PDF Máximo 1MB)

Funcionalidades clave específicas del proyecto

● Página de edición de perfil de usuario (Reclutador) : deberán crear un formulario que permita al usuario editar la información de su cuenta, y pueda completar su perfil agregando la siguiente información:

- Foto de perfil (adjunto jpg o png, Máximo 200KB)
- Título de la empresa
- Descripción de la empresa
- link sitio web de la empresa

● Página de perfil de usuario (Desarrollador) : mostrar dinámicamente el contenido del perfil del desarrollador:

- Foto de perfil
- Años de experiencia
- Estudios realizados
- Habilidades
- Tecnologías y/o lenguajes de programación
- Portafolio
- Proyectos destacados
- Sobre mi
- Certificaciones
- Idiomas
- Descarga CV

● Página de edición de perfil de usuario (Reclutador) : mostrar dinámicamente el contenido de la empresa y las vacantes disponibles de la misma:

- Foto de perfil
- Título de la empresa
- Descripción de la empresa
- link sitio web de la empresa
- Vacantes disponibles

Funcionalidades clave específicas del proyecto

● Página de Categoría: deberás crear unas categorías Deberán mostrar la información de las vacantes asociadas a la categoría escogida, con la siguiente información:

- Título de la vacante
- Años de experiencia que se requiere para la vacante
- Rango salarial de la vacante
- Ubicación.
- Botón ver más para ver el detalle de la vacante

● Página Detalle vacante con postulación: deberán mostrar el detalle de la vacante

- Título de la vacante
- Años de experiencia que se requiere para la vacante
- Rango salarial de la vacante
- Ubicación
- Descripción de la vacante
- Tecnologías o Lenguajes necesarios para aplicar a la vacante Si un desarrollador ya inició sesión deberá aparecer un botón que permita postularse a la vacante, de lo contrario deberá tener una acción para que el usuario inicie sesión para que pueda realizar su postulación. Una vez realizada la postulación, el desarrollador no podrá volverse a postular. Deberán notificar que ya realizó esta postulación.

Funcionalidades clave específicas del proyecto

● Panel de control de vacantes: mostrar la información de las vacantes registradas del reclutador:

○ Título de la vacante

○ Estado de la vacante

○ Candidatos

○ Acciones: Editar, Eliminar y ver la vacante, ver candidatos Plus (opcional): Crear una notificación toast que avise cuando un candidato se postula a la vacante.

● Registrar una vacante: Crear un formulario que registre una vacante con los siguientes campos:

○ Título de la vacante

○ Descripción del cargo

○ Ubicación: País

○ Rango salarial de la oferta

○ Tipo de Moneda

○ Años de experiencia solicitados para el cargo

○ Tecnologías requeridas para el cargo

○ Fecha final de postulación

● Ver candidatos: Crear una vista donde se muestren los candidatos que aplicaron a una oferta. plus (opcional): agregar un buscador que muestre la información filtrada de acuerdo a una coincidencia escrita por un reclutador

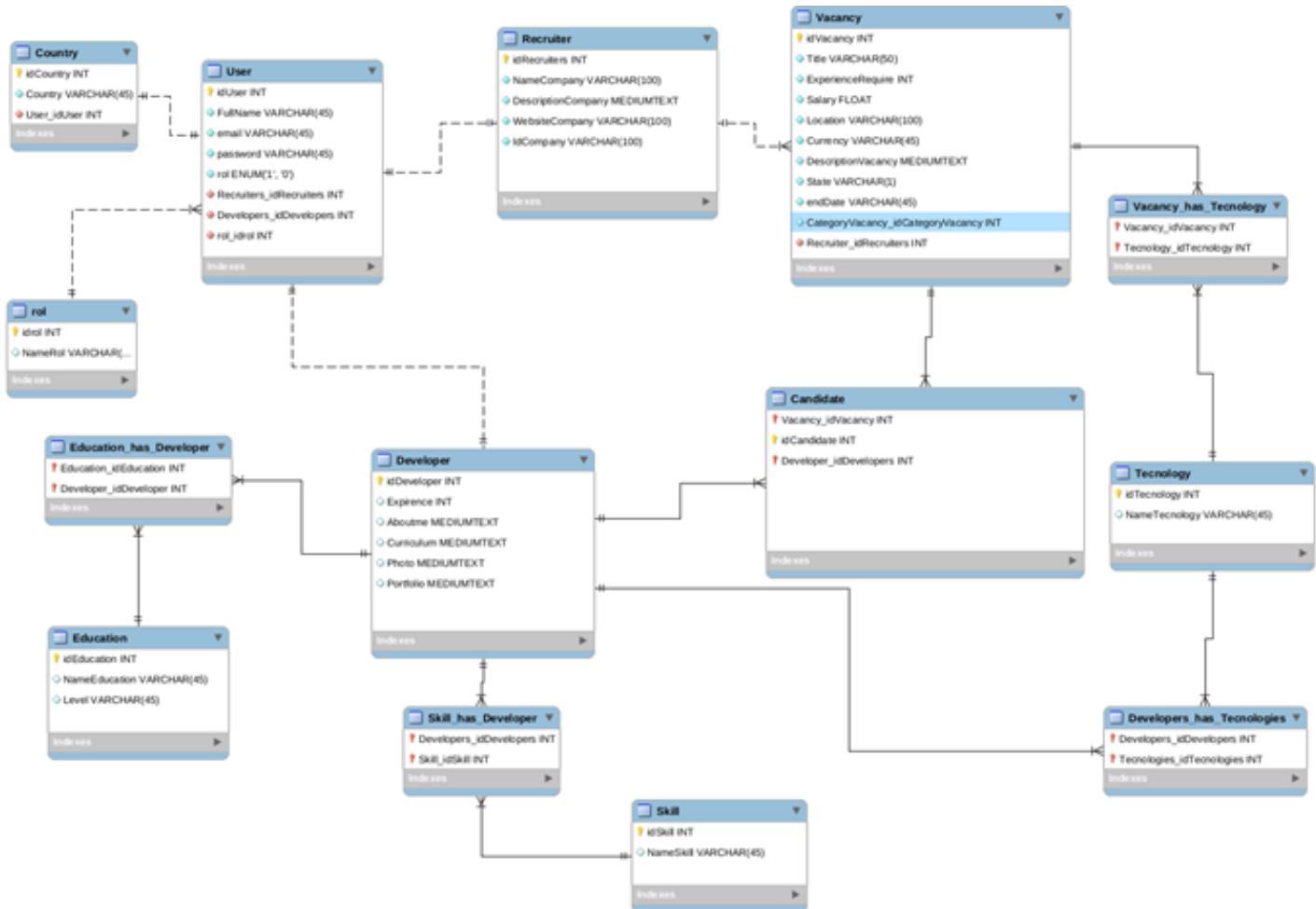
Ejecución

Bases de datos y migraciones

DOCUMENTACIÓN DEL DISEÑO DE LA BASE DE DATOS 1.

La base de datos fue diseñada en un diagrama entidad-relación, posteriormente fue pasada a un diagrama físico entidad relación usando Workbench , y finalmente se utilizó un motor de base de datos MariaDB en su versión 10.4.18-MariaDB, el consumo de los datos se hizo desde el acceso web , la manipulación de los datos se realizó utilizando el lenguaje de programación PHP en su versión 8.0.7 con el framework Laravel en su versión 4.2.5.

A continuación, se muestra el diagrama relacional de la base de datos.



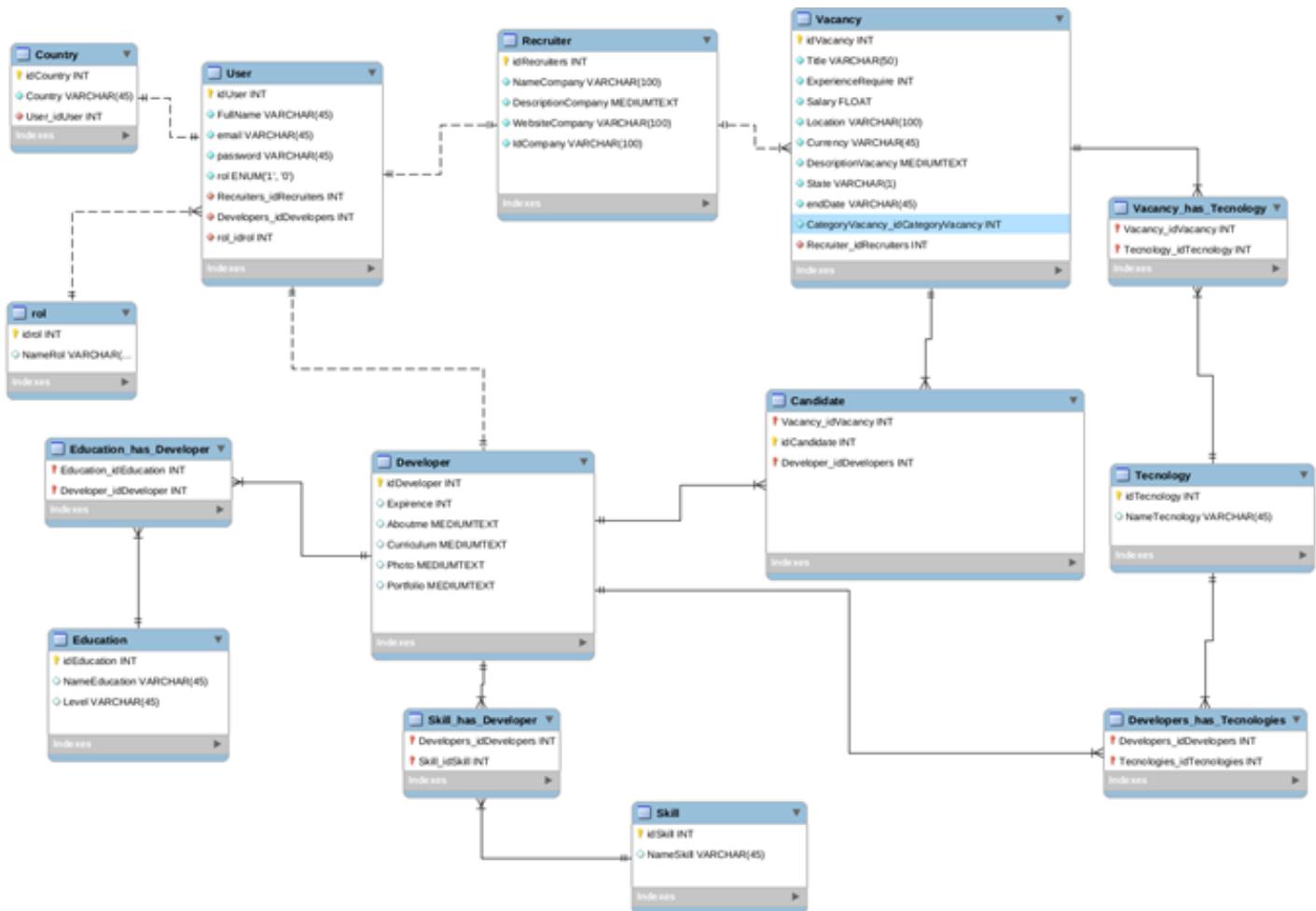
Ejecución

Bases de datos y migraciones

DOCUMENTACIÓN DEL DISEÑO DE LA BASE DE DATOS 1.

La base de datos fue diseñada en un diagrama entidad-relación, posteriormente fue pasada a un diagrama físico entidad relación usando Workbench , y finalmente se utilizó un motor de base de datos MariaDB en su versión 10.4.18-MariaDB, el consumo de los datos se hizo desde el acceso web , la manipulación de los datos se realizó utilizando el lenguaje de programación PHP en su versión 8.0.7 con el framework Laravel en su versión 4.2.5.

A continuación, se muestra el diagrama relacional de la base de datos.



Base de datos final

二

Ejecución



A screenshot of Visual Studio Code showing the configuration file `.env` for a Laravel application. The file contains environment variables such as `APP_NAME=Laravel`, `APP_ENV=local`, `APP_KEY=base64:970Ls3hoHIIkV/nq8tY346BXneOK1X6bELTftrcsH-`, `APP_DEBUG=true`, `APP_URL=http://WorkDev.test`, `LOG_CHANNEL=stack`, `LOG_LEVEL=debug`, `DB_CONNECTION=mysql`, `DB_HOST=127.0.0.1`, `DB_PORT=3306`, `DB_DATABASE=workdev`, `DB_USERNAME=root`, `DB_PASSWORD=`, `BROADCAST_DRIVER=log`, `CACHE_DRIVER=file`, `QUEUE_CONNECTION=sync`, `SESSION_DRIVER=database`, `SESSION_LIFETIME=120`, `MEMCACHED_HOST=127.0.0.1`, `REDIS_HOST=127.0.0.1`, `REDIS_PASSWORD=null`, `REDIS_PORT=6379`, `MAIL_MAILER=smtp`, `MAIL_HOST=mailhog`, `MAIL_PORT=1025`, `MAIL_USERNAME=null`, and `MAIL_PASSWORD=null`.

Ejecución Migraciones



A screenshot of Visual Studio Code showing the PHP migration file `2014_10_12_000000_create_users_table.php`. The code defines a `CreateUsersTable` class that extends `Migration`. The `up()` method creates a `users` table with columns for `id` (primary key), `name` (string), `email` (string, unique, nullable), `email_verified_at` (timestamp, nullable), `password` (string), `remember_token` (string), `current_team_id` (foreign key, nullable), and `profile_photo_path` (text, nullable). The `down()` method reverses the migrations.

```

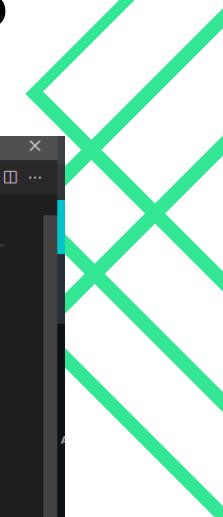
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->foreignId('current_team_id')->nullable();
            $table->text('profile_photo_path')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
    }
}

```

Ejecución Migraciones



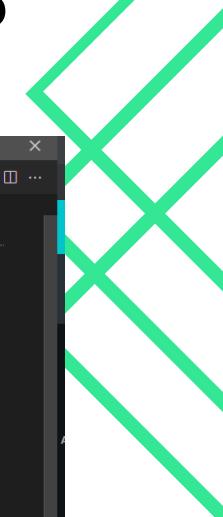
File Edit Selection View Go Run Terminal Help 2014_10_12_100000_create_password_resets_table.php - WorkDev - Visual Studio Code

EXPLORER adminlte.php 2014_10_12_100000_create_password_resets_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreatePasswordResetsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('password_resets', function (Blueprint $table) {
17             $table->string('email')->index();
18             $table->string('token');
19             $table->timestamp('created_at')->nullable();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      *
26      * @return void
27      */
28     public function down()
29     {
30         Schema::dropIfExists('password_resets');
31     }
32 }
```

In 1 Col 1 Spaces: 4 UITE:8 CRLE: PHP



File Edit Selection View Go Run Terminal Help 2014_10_12_200000_add_two_factor_columns_to_users_table.php - WorkDev - Visual Studio Code

EXPLORER adminlte.php 2014_10_12_200000_add_two_factor_columns_to_users_table.php

```

6
7 class AddTwoFactorColumnsToUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::table('users', function (Blueprint $table) {
17             $table->text('two_factor_secret')
18                 ->after('password')
19                 ->nullable();
20
21             $table->text('two_factor_recovery_codes')
22                 ->after('two_factor_secret')
23                 ->nullable();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::table('users', function (Blueprint $table) {
35             $table->dropColumn('two_factor_secret', 'two_factor_recovery_codes');
36         });
37     }
38 }
```

In 1 Col 1 Spaces: 4 UITE:8 CRLE: PHP

Ejecución Migraciones

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a tree view of the project structure under "WORKDEV". It includes "config", "database", "factories", "migrations" (containing files like "2014_10_12_000000_create_users_table.php", "2014_10_12_100000_create_password_resets_table.php", "2014_10_12_200000_add_two_fact...", "2019_08_19_000000_create_failed_jobs_table.php", etc.), "seeders", and ".gitignore".
- Editor (Center):** Displays the content of the file "2019_08_19_000000_create_failed_jobs_table.php". The code is a PHP migration class extending "Migration". It defines an "up()" method to create a table named "failed_jobs" with columns: id (string), uuid (string), connection (text), queue (text), payload (longText), exception (longText), and failed_at (timestamp). It also defines a "down()" method to reverse the migration.
- Status Bar (Bottom):** Shows the file name "2019_08_19_000000_create_failed_jobs_table.php", the current line "Line 1 Col 1", and the file type "PHP".

The screenshot shows a Visual Studio Code interface with the following details:

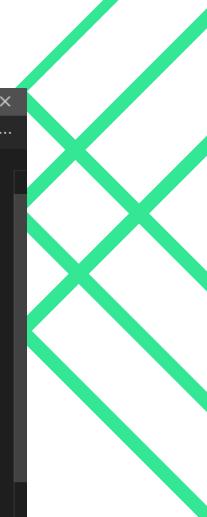
- File Explorer:** Shows the project structure under "WORKDEV".
- Editor:** Displays the file `2019_12_14_000001_create_personal_access_tokens_table.php`. The code is a PHP migration for Laravel's Blueprint API, defining a new table named `personal_access_tokens`.
- Code Content:**

```
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreatePersonalAccessTokenTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('personal_access_tokens', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->morphs('tokenable');
            $table->string('name');
            $table->string('token', 64)->unique();
            $table->text('abilities')->nullable();
            $table->timestamp('last_used_at')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('personal_access_tokens');
    }
}
```
- Status Bar:** Shows "In 17 Col 41 Spaces: 4 UTF-8 CRLF PHP".

Ejecución Migraciones



File Edit Selection View Go Run Terminal Help 2020_05_21_100000_create_teams_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... admintite.php 2020_05_21_100000_create_teams_table.php

database > migrations > 2020_05_21_100000_create_teams_table.php

```

4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateTeamsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('teams', function (Blueprint $table) {
17             $table->id();
18             $table->foreignId('user_id')->index();
19             $table->string('name');
20             $table->boolean('personal_team');
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30     public function down()
31     {
32         Schema::dropIfExists('teams');
33     }
34 }
35
36
37

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP

Juliana 0 0 ▲ 0

File Edit Selection View Go Run Terminal Help 2020_05_21_200000_create_team_user_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... admintite.php 2020_05_21_200000_create_team_user_table.php

database > migrations > 2020_05_21_200000_create_team_user_table.php

```

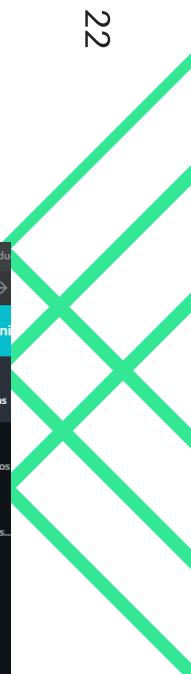
6
7 class CreateTeamUserTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('team_user', function (Blueprint $table) {
17             $table->id();
18             $table->foreignId('team_id');
19             $table->foreignId('user_id');
20             $table->string('role')->nullable();
21             $table->timestamps();
22
23             $table->unique(['team_id', 'user_id']);
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('team_user');
35     }
36 }
37

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP

Juliana 0 0 ▲ 0

Ejecución Migraciones



File Edit Selection View Go Run Terminal Help 2020_05_21_300000_create_team_invitations_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... admintite.php 2020_05_21_300000_create_team_invitations_table.php

database > migrations > 2020_05_21_300000_create_team_invitations_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateTeamInvitationsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('team_invitations', function (Blueprint $table) {
17             $table->id();
18             $table->foreignId('team_id')->constrained()->cascadeOnDelete();
19             $table->string('email');
20             $table->string('role')->nullable();
21             $table->timestamps();
22
23             $table->unique(['team_id', 'email']);
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34
35     }
36 }
```

In 1 Col 1 Spaces: 4 UITE: 8 CBLF: PHP ⌂

File Edit Selection View Go Run Terminal Help 2021_05_10_151836_create_sessions_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... admintite.php 2021_05_10_151836_create_sessions_table.php

database > migrations > 2021_05_10_151836_create_sessions_table.php

```

9 /**
10  * Run the migrations.
11  *
12  * @return void
13  */
14 public function up()
15 {
16     Schema::create('sessions', function (Blueprint $table) {
17         $table->string('id')->primary();
18         $table->foreignId('user_id')->nullable()->index();
19         $table->string('ip_address', 45)->nullable();
20         $table->text('user_agent')->nullable();
21         $table->text('payload');
22         $table->integer('last_activity')->index();
23     });
24 }
25
26 /**
27  * Reverse the migrations.
28  *
29  * @return void
30  */
31 public function down()
32 {
33     Schema::dropIfExists('sessions');
34 }
35 }
```

In 1 Col 1 Spaces: 4 UITE: 8 CBLF: PHP ⌂

Ejecución Migraciones

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "WORKDEV". The "migrations" folder contains numerous migration files, with one file named "2021_05_10_210026_create_recruiters_table.php" currently selected.
- Code Editor (Center):** Displays the content of the selected migration file. The code defines a class "CreateRecruitersTable" that extends "Migration". It includes methods for creating the "recruiters" table and defining foreign key relationships for the "user_id" column.
- Status Bar (Bottom):** Shows the current file path as "2021_05_10_210026_create_recruiters_table.php - WorkDev - Visual Studio Code", along with other standard status bar information like line and column numbers.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a tree view of files and folders. The 'migrations' folder is expanded, displaying numerous migration files such as '2014_10_12_000000_create_users...', '2014_10_12_100000_create_password...', etc., up to '2021_05_11_171933_create_permission_tables.php'. Other sections like 'config', 'database', 'factories', and 'seeders' are also visible.
- Code Editor:** The main area displays the content of the file '2021_05_11_171933_create_permission_tables.php'. The code defines several database schema elements using the Blueprint API:
 - A table for permissions: \$table->create(\$tableNames['permissions'], function (Blueprint \$table) { ... })
 - A table for roles: \$table->create(\$tableNames['roles'], function (Blueprint \$table) { ... })
 - A junction table for model_has_permissions: \$table->create(\$tableNames['model_has_permissions'], function (Blueprint \$table) { ... })
- Terminal:** At the bottom, there is a terminal window showing the command 'php artisan config:clear'.
- Status Bar:** The status bar at the bottom right shows 'Ln 1, Col 1' and other standard status indicators.

Ejecución Migraciones



File Edit Selection View Go Run Terminal Help 2021_05_11_222749_create_developers_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... adminlte.php 2021_05_11_222749_create_developers_table.php

```

<?php

namespace Illuminate\Database\Migrations;

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Migrations\Migration;

class CreateDevelopersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('developers', function (Blueprint $table) {
            $table->id();
            $table->bigInteger('user_id')->unsigned();
            $table->string('fullName');
            $table->integer('experience');
            $table->mediumText('about_me');
            $table->string('curriculum');
            $table->string('country');
            $table->string('githubProfile');
            $table->timestamps();

            $table->foreign('user_id')->references('id')->on("users")
                ->onDelete('cascade')
                ->onUpdate('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('developers');
    }
}

```

database > migrations > 2021_05_11_222749_create_developers_table.php

config factories migrations seeders .gitignore

OUTLINE TIMELINE

File Edit Selection View Go Run Terminal Help 2021_05_12_152424_create_vacancies_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... adminlte.php 2021_05_12_152424_create_vacancies_table.php

```

<?php

namespace Illuminate\Database\Migrations;

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Migrations\Migration;

class CreateVacanciesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('vacancies', function (Blueprint $table) {
            $table->id();
            $table->string('Title');
            $table->integer('ExperienceRequire');
            $table->double('Salary');
            $table->string('location');
            $table->string('currency');
            $table->longText('DescriptionVacancy');
            $table->boolean('state')->default(false);

            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('vacancies');
    }
}

```

database > migrations > 2021_05_12_152424_create_vacancies_table.php

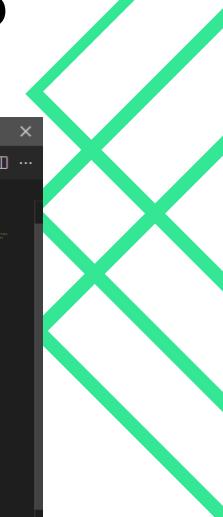
config factories migrations seeders .gitignore

OUTLINE TIMELINE

Juliana 0 0 △ 0

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP

Ejecución Migraciones



File Edit Selection View Go Run Terminal Help 2021_05_12_154002_add_recruiter_id_at_vacancy.php - WorkDev - Visual Studio Code

EXPLORER adminite.php 2021_05_12_154002_add_recruiter_id_at_vacancy.php

```

    WORKDEV migrations
      > config
      > database
        > factories
        > migrations
          2014_10_12_000000_create_users...
          2014_10_12_100000_create_passw...
          2014_10_12_200000_add_two_fact...
          2019_08_19_000000_create_failed...
          2019_12_14_000001_create_perso...
          2020_05_21_100000_create_teams...
          2020_05_21_200000_create_team...
          2020_05_21_300000_create_team...
          2021_05_10_151836_create_sessio...
          2021_05_10_210026_create_recrui...
          2021_05_11_171933_create_permi...
          2021_05_11_222749_create_devel...
          2021_05_12_152424_create_vacan...
          2021_05_12_154002_add_recruiter...
          2021_05_14_144631_create_skills...
          2021_05_14_205955_create_tecnol...
          2021_05_14_221121_create_tecno...
          2021_05_19_153946_create_educa...
          2021_05_21_224605_create_devel...
          2021_06_16_180804_create_table...
        > seeders
        > .gitignore
      > OUTLINE
      > TIMELINE
  
```

database > migrations > 2021_05_12_154002_add_recruiter_id_at_vacancy.php

```

4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class AddRecruiterIdAtVacancy extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::table('vacancies', function (Blueprint $table){
17
18             $table->bigInteger('recruiter_id')->unsigned()->nullable()->after('id');
19
20             $table->foreign('recruiter_id')->references('id')->on("recruiters")
21             ->onDelete('cascade')
22             ->onUpdate('cascade');
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31     public function down()
32     {
33         //
34     }
35 }
  
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP

File Edit Selection View Go Run Terminal Help 2021_05_14_144631_create_skills_table.php - WorkDev - Visual Studio Code

EXPLORER adminite.php 2021_05_14_144631_create_skills_table.php

```

    WORKDEV migrations
      > config
      > database
        > factories
        > migrations
          2014_10_12_000000_create_users...
          2014_10_12_100000_create_passw...
          2014_10_12_200000_add_two_fact...
          2019_08_19_000000_create_failed...
          2019_12_14_000001_create_perso...
          2020_05_21_100000_create_teams...
          2020_05_21_200000_create_team...
          2020_05_21_300000_create_team...
          2021_05_10_151836_create_sessio...
          2021_05_10_210026_create_recrui...
          2021_05_11_171933_create_permi...
          2021_05_11_222749_create_devel...
          2021_05_12_152424_create_vacan...
          2021_05_12_154002_add_recruiter...
          2021_05_14_144631_create_skills...
          2021_05_14_205955_create_tecnol...
          2021_05_14_221121_create_tecno...
          2021_05_19_153946_create_educa...
          2021_05_21_224605_create_devel...
          2021_06_16_180804_create_table...
        > seeders
        > .gitignore
      > OUTLINE
      > TIMELINE
  
```

database > migrations > 2021_05_14_144631_create_skills_table.php

```

12     * @return void
13     */
14     public function up()
15     {
16         //crea tabla skills
17         Schema::create('skills', function (Blueprint $table) {
18             $table->bigIncrements('id');
19             $table->timestamps();
20             $table->string('skillName');
21         });
22         //crea tabla intermedia desarrollador tiene habilidades
23         Schema::create('developer_skill',function (Blueprint $table){
24
25             $table->unsignedBigInteger('skill_id');
26             $table->unsignedBigInteger('developer_id');
27
28             $table->timestamps();
29
30             $table->foreign('skill_id')
31                 ->references('id')
32                 ->on('skills')
33                 ->onDelete('cascade');
34
35             $table->foreign('developer_id')
36                 ->references('user_id')
37                 ->on('developers')
38                 ->onDelete('cascade');
39         });
40     }
41
42     /**
43      * Reverse the migrations.
44      */
45
  
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP

Ejecución Migraciones



File Edit Selection View Go Run Terminal Help 2021_05_14_205955_create_tecnologies_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... admintite.php 2021_05_14_205955_create_tecnologies_table.php

> config > database > migrations > 2021_05_14_205955_create_tecnologies_table.php

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateTecnologiesTable extends Migration
8 {
9
10    /**
11     * Run the migrations.
12     *
13     * @return void
14     */
15    public function up()
16    {
17        Schema::create('tecnologies', function (Blueprint $table) {
18            $table->id();
19            $table->string("tecnico");
20            $table->timestamps();
21        });
22    }
23
24    /**
25     * Reverse the migrations.
26     *
27     * @return void
28     */
29    public function down()
30    {
31        Schema::dropIfExists('tecnologies');
32    }
33

```

> OUTLINE > TIMELINE

Juliana 0 △ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡



File Edit Selection View Go Run Terminal Help 2021_05_14_221121_create_tecno_vacancy_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... admintite.php 2021_05_14_221121_create_tecno_vacancy_table.php

> config > database > factories > migrations > 2021_05_14_221121_create_tecno_vacancy_table.php

```

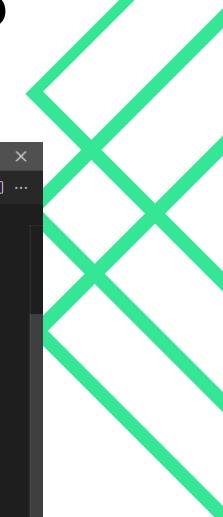
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateTecnoVacancyTable extends Migration
8 {
9
10    /**
11     * Run the migrations.
12     *
13     * @return void
14     */
15    public function up()
16    {
17        Schema::create('technology_vacancy', function (Blueprint $table) {
18            $table->id();
19            $table->bigInteger('technology_id')->unsigned();
20            $table->bigInteger('vacancy_id')->unsigned();
21
22            $table->timestamps();
23
24            $table->foreign('technology_id')->references('id')->on("tecnoologies")
25                ->onDelete('cascade')
26                ->onUpdate('cascade');
27
28            $table->foreign('vacancy_id')->references('id')->on("vacancies")
29                ->onDelete('cascade')
30                ->onUpdate('cascade');
31        });
32    }
33

```

> seeders > .gitignore > OUTLINE > TIMELINE

Juliana 0 △ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡

Ejecución Migraciones



File Edit Selection View Go Run Terminal Help 2021_05_19_153946_create_education_table.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... adminlte.php 2021_05_19_153946_create_education_table.php

database > migrations > 2021_05_19_153946_create_education_table.php

```

14 public function up()
15 {
16     //creates education table
17     Schema::create('education', function (Blueprint $table) {
18         $table->id();
19         $table->timestamps();
20         $table->string('nameEducation');
21         $table->string('level');
22     });
23     //crea tabla intermedia entre desarrollador y educación
24     Schema::create('developer_education', function(Blueprint $table){
25         $table->unsignedBigInteger('education_id');
26         $table->unsignedBigInteger('developer_id');
27
28         $table->timestamps();
29
30         $table->primary(['education_id', 'developer_id']);
31
32         $table->foreign('education_id')
33             ->references('id')
34             ->on('education')
35             ->onDelete('cascade');
36
37         $table->foreign('developer_id')
38             ->references('user_id')
39             ->on('developers')
40             ->onDelete('cascade')
41             ->onUpdate('cascade');
42     });
43 });
44 */
45

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡ 🔍

File Edit Selection View Go Run Terminal Help 2021_05_21_224605_create_developer_tecnology.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ... adminlte.php 2021_05_21_224605_create_developer_tecnology.php

database > migrations > 2021_05_21_224605_create_developer_tecnology.php

```

4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateDeveloperTecnology extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('developer_tecnology', function (Blueprint $table) {
17             $table->id();
18             $table->timestamps();
19
20             $table->bigInteger('tecnology_id')->unsigned();
21             $table->bigInteger('developer_id')->unsigned();
22
23             $table->foreign('tecnology_id')->references('id')->on("tecnoologies")
24                 ->onDelete('cascade')
25                 ->onUpdate('cascade');
26
27             $table->foreign('developer_id')->references('user_id')->on("developers")
28                 ->onDelete('cascade')
29                 ->onUpdate('cascade');
30         });
31     }
32
33     /**
34      * Reverse the migrations.
35      */

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡ 🔍

Ejecución Migraciones

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a project structure under "WORKDEV". The "migrations" folder contains numerous migration files, with one file highlighted: "2021_06_16_180804_create_table_developer_vacancy.php". Other visible files include "config", "database", "factories", "seeders", and ".gitignore".
- Editor (Center):** Displays the PHP code for the migration file. The code defines a "developer_vacancy" table with columns: "vacancy_id" (unsigned big integer), "developer_id" (unsigned big integer), and timestamps. It also defines foreign keys for "vacancy_id" referencing "vacancies" and "developer_id" referencing "developers".
- Terminal (Bottom Left):** Shows the command "Julian" followed by several small icons.
- Status Bar (Bottom Right):** Shows "Ln 1, Col 1", "Spaces: 4", "UTF-8", "CRLF", and "PHP".

```
database > migrations > 2021_06_16_180804_create_table_developer_vacancy.php
```

```
/*
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('developer_vacancy', function (Blueprint $table) {

        $table->unsignedBigInteger('vacancy_id');
        $table->unsignedBigInteger('developer_id');

        $table->timestamps();

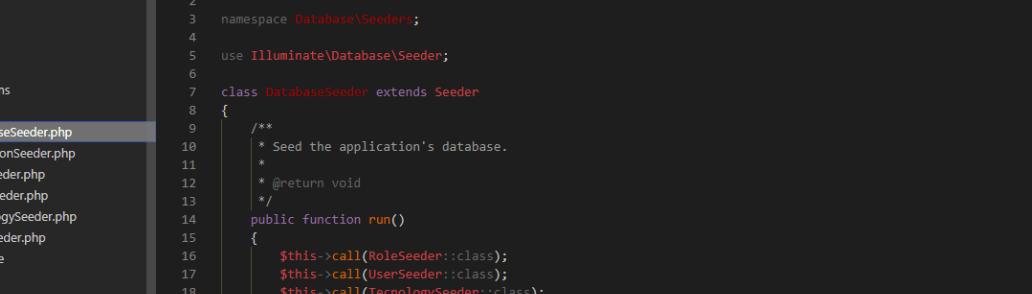
        $table->primary(['vacancy_id', 'developer_id']);

        $table->foreign('vacancy_id')->references('id')->on("vacancies")
            ->onDelete('cascade')
            ->onUpdate('cascade');

        $table->foreign("developer_id")->references('user_id')->on("developers")
            ->onDelete('cascade')
            ->onUpdate('cascade');
    });
}

/*
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
```

Ejecución SEDDERS



```
File Edit Selection View Go Run Terminal Help DatabaseSeeder.php - WorkDev - Visual Studio Code

EXPLORER ... adminlte.php DatabaseSeeder.php x
WORKDEV database > seeders > DatabaseSeeder.php
  1 <?php
  2
  3 namespace Database\Seeders;
  4
  5 use Illuminate\Database\Seeder;
  6
  7 class DatabaseSeeder extends Seeder
  8 {
  9
 10     /**
 11      * Seed the application's database.
 12      *
 13      * @return void
 14      */
 15     public function run()
 16     {
 17         $this->call(RoleSeeder::class);
 18         $this->call(UserSeeder::class);
 19         $this->call(TechnologySeeder::class);
 20         $this->call(SkillsSeeder::class);
 21         $this->call(EducationSeeder::class);
 22     }
 23 }
```

Ejecución SEDDERS



File Edit Selection View Go Run Terminal Help

EducationSeeder.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV adminite.php EducationSeeder.php

```
database > seeders > EducationSeeder.php
```

```
10 /**
11  * Run the database seeds.
12  */
13 /**
14  * @return void
15 */
16 public function run()
17 {
18     //crea un array con las carreras y los niveles
19     $careers = array(
20         "Administración y Negocios",
21         "Arquitectura",
22         "Artes Visuales",
23         "Ciencia",
24         "Ciencias Ambientales",
25         "Ciencias de la Salud",
26         "Ciencias Económicas",
27         "Ciencias Políticas",
28         "Ciencias Sociales",
29         "Comunicación Social",
30         "Derecho",
31         "Diseño Gráfico",
32         "Diseño Industrial",
33         "Diseño UI/UX",
34         "Ingeniería Civil",
35         "Ingeniería Eléctrica",
36         "Ingeniería Electrónica",
37         "Ingeniería Industrial",
38         "Ingeniería Mecánica",
39         "Ingeniería Mecatrónica",
40         "Ingeniería Química",
41         "Ingeniería Sistemas",
42         "Marketing y Publicidad"
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡



File Edit Selection View Go Run Terminal Help

RoleSeeder.php - WorkDev - Visual Studio Code

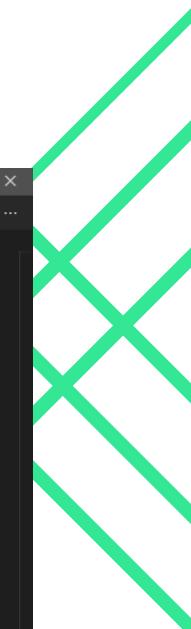
EXPLORER WORKDEV adminite.php RoleSeeder.php

```
database > seeders > RoleSeeder.php
```

```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6 use Spatie\Permission\Models\Role;
7 use Spatie\Permission\Models\Permission;
8
9 class RoleSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     /**
15      * @return void
16     */
17     public function run()
18     {
19         //creacion de los roles
20         $roleAdmin = Role::create(['name'=>"Admin"]);
21         $roleRec = Role::create(['name'=>'Recruiter']);
22         $roleDev = Role::create(['name'=>'Developer']);
23
24         //creación del permiso y asignarlo a un rol
25         //Permission::create(['name'=>'companydata'])->assignRole($roleAdmin);
26         //creación del permiso y asignación a varios roles
27         Permission::create(['name'=>'companydata'])->syncRoles([$roleAdmin, $roleRec]);
28         Permission::create(['name'=>'developerdata'])->syncRoles([$roleAdmin, $roleDev]);
```

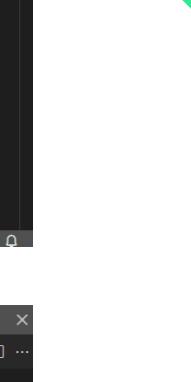
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡

Ejecución SEDDERS



Screenshot of Visual Studio Code showing the code editor with the file `SkillsSeeder.php` open. The code defines a `SkillsSeeder` class that runs database seeds for skills like teamwork, leadership, creativity, innovation, critical thinking, problem-solving, effective communication, adaptability, autonomy, collaboration, emotional intelligence, and initiative.

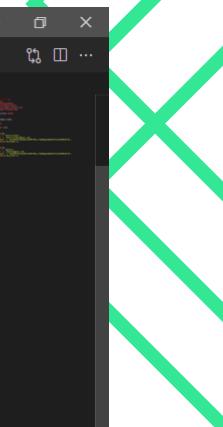
```
<?php  
namespace Database\Seeders;  
  
use Illuminate\Database\Seeder;  
use App\Models\Skill;  
use App\Models\Developer;  
  
class SkillsSeeder extends Seeder  
{  
    /**  
     * Run the database seeds.  
     *  
     * @return void  
     */  
    public function run()  
    {  
        $skills = [  
            "trabajo en equipo",  
            "liderazgo",  
            "creatividad",  
            "innovación",  
            "pensamiento crítico",  
            "resolución de problemas",  
            "comunicación efectiva",  
            "adaptabilidad",  
            "autonomía",  
            "colaboración",  
            "inteligencia emocional",  
            "iniciativa"  
        ];  
    }  
}  
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP
```



Screenshot of Visual Studio Code showing the code editor with the file `TecnologySeeder.php` open. The code defines a `TecnologySeeder` class that runs database seeds for various technologies including Java, Javascript, Python, C++, C, SQL, Mysql, AWK, C#, R, TypeScript, and PHP.

```
<?php  
namespace Database\Seeders;  
  
use Illuminate\Database\Seeder;  
use App\Models\Tecnology;  
  
class TecnologySeeder extends Seeder  
{  
    /**  
     * Run the database seeds.  
     *  
     * @return void  
     */  
    public function run()  
    {  
        $tecnologie = [  
            "Java",  
            "Javascript",  
            "Python",  
            "C++",  
            "C",  
            "SQL",  
            "Mysql",  
            "AWK",  
            "C#",  
            "R",  
            "TypeScript",  
            "PHP"  
        ];  
        foreach($tecnologie as $tecn){  
            Tecnology::create([  
                'tecn' => $tecn  
            ]);  
        }  
    }  
}  
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP
```

Ejecución SEDDERS



A screenshot of Visual Studio Code showing the code editor and file explorer. The file explorer on the left shows a project structure under 'WORKDEV' with files like DatabaseSeeder.php, EducationSeeder.php, RoleSeeder.php, SkillsSeeder.php, TechnologySeeder.php, and UserSeeder.php. The UserSeeder.php file is selected and open in the editor. The code in UserSeeder.php is as follows:

```
9  class UserSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15     */
16    public function run()
17    {
18        //
19        User::create([
20            'name' => 'desarrollador',
21            'email' => 'desarrollador@gmail.com',
22            'password' => '$2y$10$JY4JeoZZ9sjSCWHff05jy.YEq90gqukBwWH12RT2nod8oRMJ83TnK',
23        ])->assignRole('Developer');
24
25
26
27
28        User::create([
29            'name' => 'empresa',
30            'email' => 'empresa@gmail.com',
31            'password' => '$2y$10$JY4JeoZZ9sjSCWHff05jy.YEq90gqukBwWH12RT2nod8oRMJ83TnK',
32        ])->assignRole('Recruiter');
33
34
35
36    }
37}
```

The status bar at the bottom indicates the file is 'UserSeeder.php - WorkDev - Visual Studio Code', has 1 line, 1 column, 4 spaces, is in UTF-8 encoding, and is a PHP file.

Ejecución Controladores

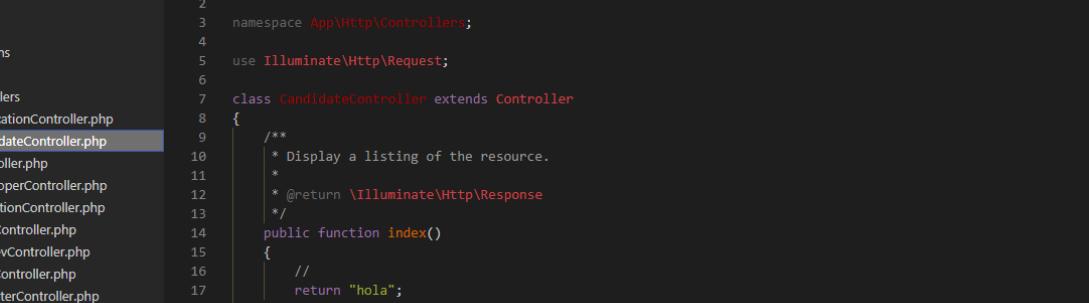
```
File Edit Selection View Go Run Terminal Help ApplicationController.php - WorkDev - Visual Studio Code

EXPLORER ... ApplicationController.php x
WORKDEV app > Http > Controllers > ApplicationController.php

app 17 * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //obtiene el id del usuario actual
        $userId = Auth::id();
        //verifica si el usuario ya tiene sus datos llenos
        $id_developer = DB::table('users')
            ->join('developers', 'users.id', '=', 'developers.user_id')
            ->where('developers.user_id', '=', $userId)
            ->select('developers.id')
            ->get();
        if(empty($id_developer[0]->id)){
            //get actual user applications
            $userApplication = DB::table('vacancies')
                ->join('developer_vacancy', 'vacancies.id', '=', 'developer_vacancy.vacancy_id')
                ->join('recruiters', 'vacancies.recruitier_id', '=', 'vacancies.recruitier_id')
                ->join('users', 'recruiters.user_id', '=', 'users.id')
                ->where('developer_vacancy.developer_id', '=', $userId)
                ->where('state', '=', 1)
                ->select('vacancies.*', 'recruiters.*', 'users.profile_photo_path')
                ->get();
        }
        else{
            $userApplication="mensaje de error";
        }
    }

    return view('developer.applications', ['userApplication' => $userApplication]);
}

Juliana 47 Spaces: 4 48 UTF-8 CRLF PHP
```



The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** CandidateController.php - WorkDev - Visual Studio Code.
- Explorer Bar (Left):** Shows the project structure under WORKDEV:
 - app
 - > Actions
 - > Console
 - > Exceptions
 - Http
 - > Controllers
 - ApplicationController.php
 - CandidateController.php
 - DeveloperController.php
 - EducationController.php
 - FilterController.php
 - JobDevController.php
 - PageController.php
 - RecruiterController.php
 - SkillController.php
 - TecnologyController.php
 - VacancyController.php
 - > Livewire
 - > Middleware
 - Kernel.php
 - > Models
 - > Providers
 - > View
 - > bootstrap
 - > config
 - > OUTLINE
 - > TIMELINE
- Editor Area (Center):** Displays the PHP code for CandidateController.php.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class CandidateController extends Controller
8 {
9
10    /**
11     * Display a listing of the resource.
12     *
13     * @return \Illuminate\Http\Response
14     */
15    public function index()
16    {
17        //
18        return "hola";
19    }
20
21    /**
22     * Show the form for creating a new resource.
23     *
24     * @return \Illuminate\Http\Response
25     */
26    public function create()
27    {
28        //
29    }
30
31    /**
32     * Store a newly created resource in storage.
33     *
34     * @param \Illuminate\Http\Request $request
35     */
36}
```
- Right Side Panel:** Shows the PHP Intelephense code completion feature with suggestions for the \$request parameter.
- Bottom Status Bar:** Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | PHP | ⚡

Ejecución Controladores

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** DeveloperController.php - WorkDev - Visual Studio Code.
- Left Sidebar (Explorer):** WORKDEV folder expanded, showing subfolders like app, Actions, Console, Exceptions, Http, and Controllers. Under Controllers, several files are listed: ApplicationController.php, CandidateController.php, Controller.php, DeveloperController.php (highlighted), EducationController.php, FilterController.php, JobDevController.php, PageController.php, RecruiterController.php, SkillController.php, TechnologyController.php, and VacancyController.php. Below these are Livewire, Middleware, Kernel.php, Models, Providers, View, bootstrap, config, OUTLINE, and TIMELINE.
- Central Area:** The code editor displays the contents of the DeveloperController.php file. The code handles file uploads, initializes a developer object with user data, saves it to the database, and returns a redirect response.
- Right Sidebar:** Includes a Git status bar, a PHP Intelephense status bar, and a Taskbar with various icons.

```
DeveloperController.php

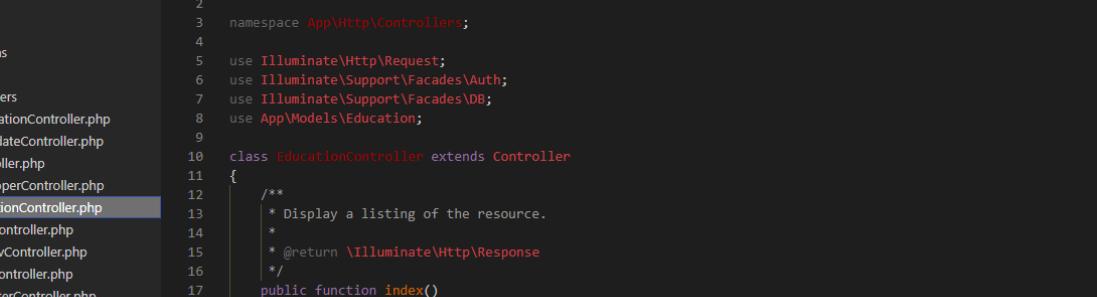
43     $file=$request->file('curriculum');
44     //nombra el archivo con la fecha la hora y la extension
45     $name = "pdf_".time().".".file->guessExtension();
46     //asigna la ruta para guardarla
47     $route = public_path("curriculums/".$name);
48     //guarda el archivo en la ruta especificada
49     copy($file, $route);

50
51
52     $dataDev = new Developer();
53     $data = $request->user();
54     $dataDev->user_id= $data['id'];
55     $dataDev->fullName = $request->get('fullName');
56     $dataDev->experience = $request->get('experience');
57     $dataDev->about_me = $request->get('about_me');
58     $dataDev->country = $request->get('country');
59     $dataDev->githubProfile = $request->get('github');
60     $dataDev->curriculum = "curriculums/".$name;

61
62     $dataDev->save();
63     $developer = Developer::where('user_id', $data['id'])->first();

64
65
66
67
68
69
70
71
72
73
74
75

}
/** 
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
```

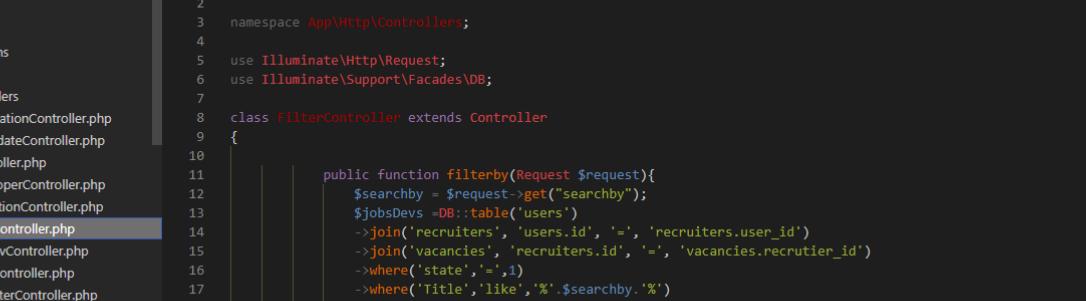


```
EducationController.php

app > Http > Controllers > EducationController.php

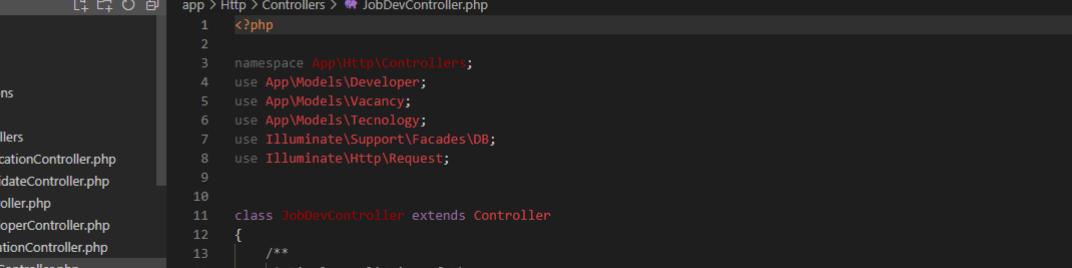
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Auth;
7 use Illuminate\Support\Facades\DB;
8 use App\Models\Education;
9
10 class EducationController extends Controller
11 {
12     /**
13      * Display a listing of the resource.
14      *
15      * @return \Illuminate\Http\Response
16      */
17     public function index()
18     {
19         $education = array(
20             "Administración y Negocios",
21             "Arquitectura",
22             "Artes Visuales",
23             "Ciencia",
24             "Ciencias Ambientales",
25             "Ciencias de la Salud",
26             "Ciencias Económicas",
27             "Ciencias Políticas",
28             "Ciencias Sociales",
29             "Comunicación Social",
30             "Derecho",
31             "Diseño Gráfico",
32             "Diseño Industrial",
33             "Diseño UI/UX"
```

Ejecución Controladores



```
File Edit Selection View Go Run Terminal Help FilterController.php - WorkDev - Visual Studio Code

EXPLORER FilterController.php x
WORKDEV app > Http > Controllers > FilterController.php
app > Http > Controllers > FilterController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7
8 class FilterController extends Controller
9 {
10
11     public function filterby(Request $request){
12         $searchby = $request->get("searchby");
13         $jobsDevs =DB::table('users')
14             ->join('recruiters', 'users.id', '=', 'recruiters.user_id')
15             ->join('vacancies', 'recruiters.id', '=', 'vacancies.recruiter_id')
16             ->where('state', '=', 1)
17             ->where('Title', 'like', '%' . $searchby . '%')
18             ->orwhere('NameCompany', 'like', '%' . $searchby . '%')
19             ->orderBy('vacancies.created_at', 'DESC')
20             ->select('users.*', 'recruiters.*','vacancies.*')
21             ->get();
22         return view("jobDev.index", ['jobsDevs'=>$jobsDevs]);
23     }
24 }
25
26 }
```



```
File Edit Selection View Go Run Terminal Help JobDevController.php - WorkDev - Visual Studio Code

EXPLORER ... WORKDEV app > Http > Controllers > JobDevController.php
1 <?php
2
3 namespace App\Http\Controllers;
4 use App\Models\Developer;
5 use App\Models\Vacancy;
6 use App\Models\Technology;
7 use Illuminate\Support\Facades\DB;
8 use Illuminate\Http\Request;
9
10
11 class JobDevController extends Controller
12 {
13     /**
14      * Display a listing of the resource.
15      *
16      * @return \Illuminate\Http\Response
17      */
18     public function index()
19     {
20         $jobsDevs = DB::table('users')
21             ->join('recruiters', 'users.id', '=', 'recruiters.user_id')
22             ->join('vacancies', 'recruiters.id', '=', 'vacancies.recruiter_id')
23             ->where('state', '=', 1)
24             ->orderBy('vacancies.created_at', 'DESC')
25             ->select('users.*', 'recruiters.*', 'vacancies.*')
26             ->get();
27         return view("jobDev.index", ['jobsDevs' => $jobsDevs]);
28     }
29
30
31
32
33     public function jobdetail($title, $id)
34 }
```

Ejecución Controladores

File Edit Selection View Go Run Terminal Help PageController.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV PageController.php

```

12 class PageController extends Controller
13 {
14     public function home(){
15         return view("welcome");
16     }
17
18     public function nosotros(){
19         return view('nosotros');
20     }
21
22     public function companydata(){
23         return view('companydata');
24     }
25
26     public function developerdata(){
27         $id = Auth::id();
28         $developer = Developer::where('user_id', $id)->first();
29         // dd($developer);
30         return view('developer.developerData', [
31             'developer' => $developer]);
32     }
33
34     public function editDeveloper(){
35         $id = Auth::id();
36         $developer = $developer = Developer::where('user_id', $id)->first();
37         // dd($developer);
38         return view('developer.editDeveloper', [
39             'developer' => $developer]);
40     }
41
42     public function vacancy(Vacancy $vacancy){
43

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚙

File Edit Selection View Go Run Terminal Help RecruiterController.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV RecruiterController.php

```

12 class RecruiterController extends Controller
13 {
14     /**
15      * Display a listing of the resource.
16      *
17      * @return \Illuminate\Http\Response
18     */
19     public function index(Request $request)
20     {
21
22         $data=$request->user();
23         $users = DB::table('users')
24             ->join('recruiters', 'users.id', '=', 'recruiters.user_id')
25             ->select('recruiters.*')
26             ->where('recruiters.user_id', '=', $data['id'])
27             ->get();
28
29         return view("Recruiter.companydata", ['users'=> $users]);
30     }
31
32     /**
33      * Show the form for creating a new resource.
34      *
35      * @return \Illuminate\Http\Response
36     */
37     public function create()
38     {
39         //
40     }
41
42     /**
43

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚙

Ejecución Controladores

SkillController.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER WORKDEV ...
app > Http > Controllers > SkillController.php
SkillController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Auth;
7 use Illuminate\Support\Facades\DB;
8 use App\Models\Skill;
9
10 class SkillController extends Controller
11 {
12     /**
13      * Display a listing of the resource.
14      *
15      * @return \Illuminate\Http\Response
16     */
17     public function index()
18     {
19         //obtiene el id del usuario actual
20         $userId = Auth::id();
21         //verifica si el usuario ya tiene sus datos llenos
22         $id_developer = DB::table('users')
23             ->join('developers', 'users.id', '=', 'developers.user_id')
24             ->where('developers.user_id', '=', $userId)
25             ->select('developers.id')
26             ->get();
27         if(empty($id_developer[0]->id)){
28             //obtiene las habilidades del usuario actual
29             $userSkills = DB::table('skills')
30                 ->join('developer_skill', 'skills.id', '=', 'developer_skill.skill_id')
31                 ->where('developer_skill.developer_id', '=', $userId)
32                 ->select('skills.skillName', 'developer_skill.skill_id')
33                 ->get();
34         }
35     }
36 }

```

TecnologyController.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER WORKDEV ...
app > Http > Controllers > TecnologyController.php
TecnologyController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Auth;
7 use Illuminate\Support\Facades\DB;
8 use App\Models\Tecnology;
9
10 class TecnologyController extends Controller
11 {
12     /**
13      * Display a listing of the resource.
14      *
15      * @return \Illuminate\Http\Response
16     */
17     public function index()
18     {
19         //obtiene el id del usuario actual
20         $userId = Auth::id();
21         //obtiene la lista de tecnologias
22         $tecnologies = DB::table('tecnologies')
23             ->select('tecnologies.tecno', 'tecnologies.id')
24             ->get();
25         //verifica si el usuario ya tiene sus datos llenos
26         $id_developer = DB::table('users')
27             ->join('developers', 'users.id', '=', 'developers.user_id')
28             ->where('developers.user_id', '=', $userId)
29             ->select('developers.id')
30             ->get();
31         if(empty($id_developer[0]->id)){
32             //obtiene las tecnologias del usuario actual
33         }
34     }
35 }

```

Ejecución Controladores

VacancyController.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help VacancyController.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ...
app > Http > Controllers > VacancyController.php
127
128     public function edit($id)
129     {
130
131         $technologies = DB::table('technologies')
132             ->select('technologies.tecno', 'technologies.id')
133             ->get();
134
135         $vacante= Vacancy::find($id);
136
137         $userTecno = DB::table('technologies')
138             ->join('technology_vacancy', 'technologies.id','=', 'technology_vacancy.technology_id')
139             ->where('technology_vacancy.vacancy_id', '=', $id)
140             ->select('technologies.tecno', 'technologies.id')
141             ->get();
142
143
144         return view('Vacancy.edit',[ 'vacante'=>$vacante, 'userTecno'=>$userTecno , 'technologies'=>$technologies]);
145     }
146
147
148 /**
149 * Update the specified resource in storage.
150 *
151 * @param \Illuminate\Http\Request $request
152 * @param int $id
153 * @return \Illuminate\Http\Response
154 */
155 public function update(Request $request, $id)
156 {
157     $update_vacancy = Vacancy::find($id);
158     $update_vacancy->Title= $request->get('title');
159     $update_vacancy->ExperienceRequire= $request->get('experience');
}

```

JobList.php - WorkDev - Visual Studio Code

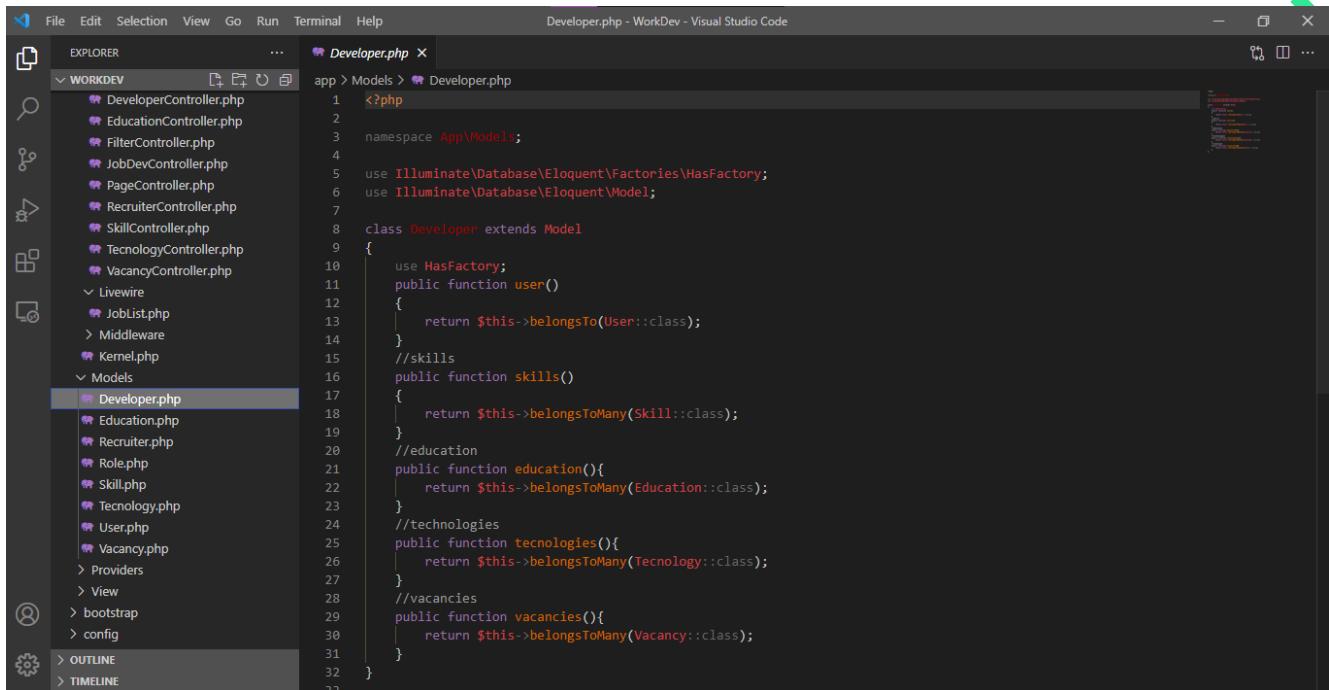
```

File Edit Selection View Go Run Terminal Help JobList.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV ...
app > Http > Livewire > JobList.php
1 <?php
2
3 namespace App\Http\Livewire;
4
5 use Livewire\Component;
6 use App\Models\Vacancy;
7 use Illuminate\Support\Facades\DB;
8
9
10 class JobList extends Component
11 {
12     public function render()
13     {
14         return view('livewire.job-list',[ 'jobs'=>DB::table('users')
15             ->join('recruiters', 'users.id', '=', 'recruiters.user_id')
16             ->join('vacancies', 'recruiters.id', '=', 'vacancies.recruiter_id')
17             ->select('users.*', 'recruiters.*','vacancies.*')
18             ->skip(0)
19             ->take(9)
20             ->get(),
21
22         'tecno'=> DB::table('technologies')
23             ->join('technology_vacancy', 'technologies.id','=', 'technology_vacancy.technology_id')
24             ->select('technologies.tecno', 'technologies.id')
25             ->get()]);
26     }
}

```

Ejecución Modelos

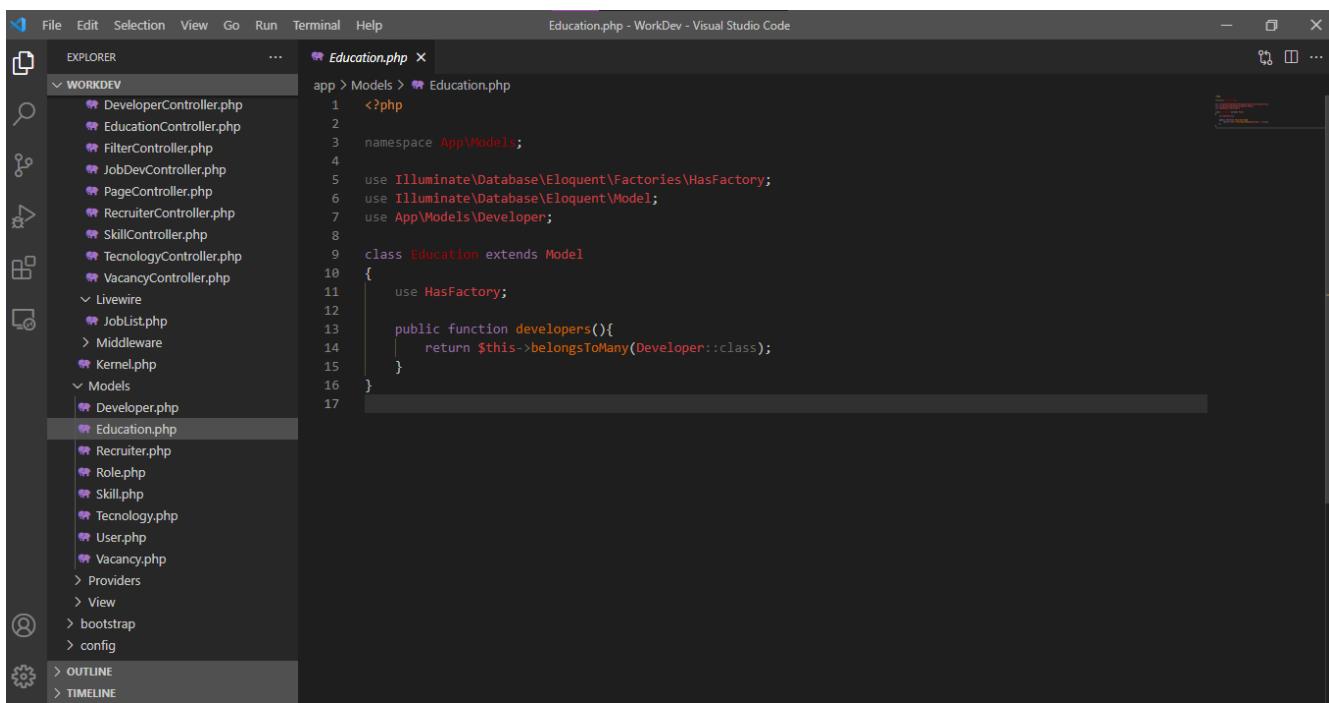


Developer.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER Developer.php ...
WORKDEV
DeveloperController.php
EducationController.php
FilterController.php
JobDevController.php
PageController.php
RecruiterController.php
SkillController.php
TecnologyController.php
VacancyController.php
Livewire
JobList.php
> Middleware
Kernel.php
Models
Developer.php
Education.php
Recruiter.php
Role.php
Skill.php
Tecnology.php
User.php
Vacancy.php
> Providers
> View
> bootstrap
> config
> OUTLINE
> TIMELINE
app > Models > Developer.php
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Developer extends Model
{
    use HasFactory;
    public function user()
    {
        return $this->belongsTo(User::class);
    }
    //skills
    public function skills()
    {
        return $this->belongsToMany(Skill::class);
    }
    //education
    public function education(){
        return $this->belongsToMany(Education::class);
    }
    //technologies
    public function tecnologies(){
        return $this->belongsToMany(Tecnology::class);
    }
    //vacancies
    public function vacancies(){
        return $this->belongsToMany(Vacancy::class);
    }
}

```



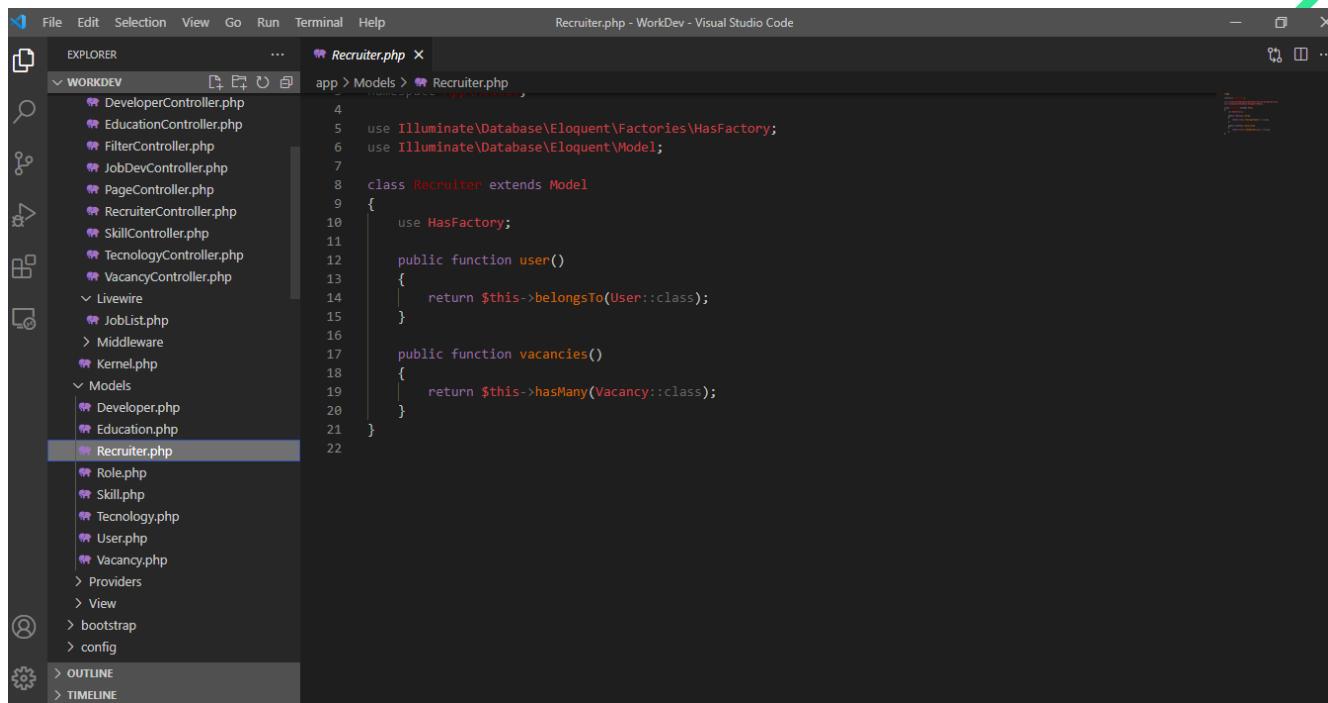
Education.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER Education.php ...
WORKDEV
DeveloperController.php
EducationController.php
FilterController.php
JobDevController.php
PageController.php
RecruiterController.php
SkillController.php
TecnologyController.php
VacancyController.php
Livewire
JobList.php
> Middleware
Kernel.php
Models
Developer.php
Education.php
Recruiter.php
Role.php
Skill.php
Tecnology.php
User.php
Vacancy.php
> Providers
> View
> bootstrap
> config
> OUTLINE
> TIMELINE
app > Models > Education.php
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use App\Models\Developer;
8
9 class Education extends Model
{
    use HasFactory;
    public function developers(){
        return $this->hasMany(Developer::class);
    }
}

```

Ejecución Modelos



File Edit Selection View Go Run Terminal Help

RECRUITER.PHP - WORKDEV - Visual Studio Code

EXPLORER ... RECRUITER.PHP

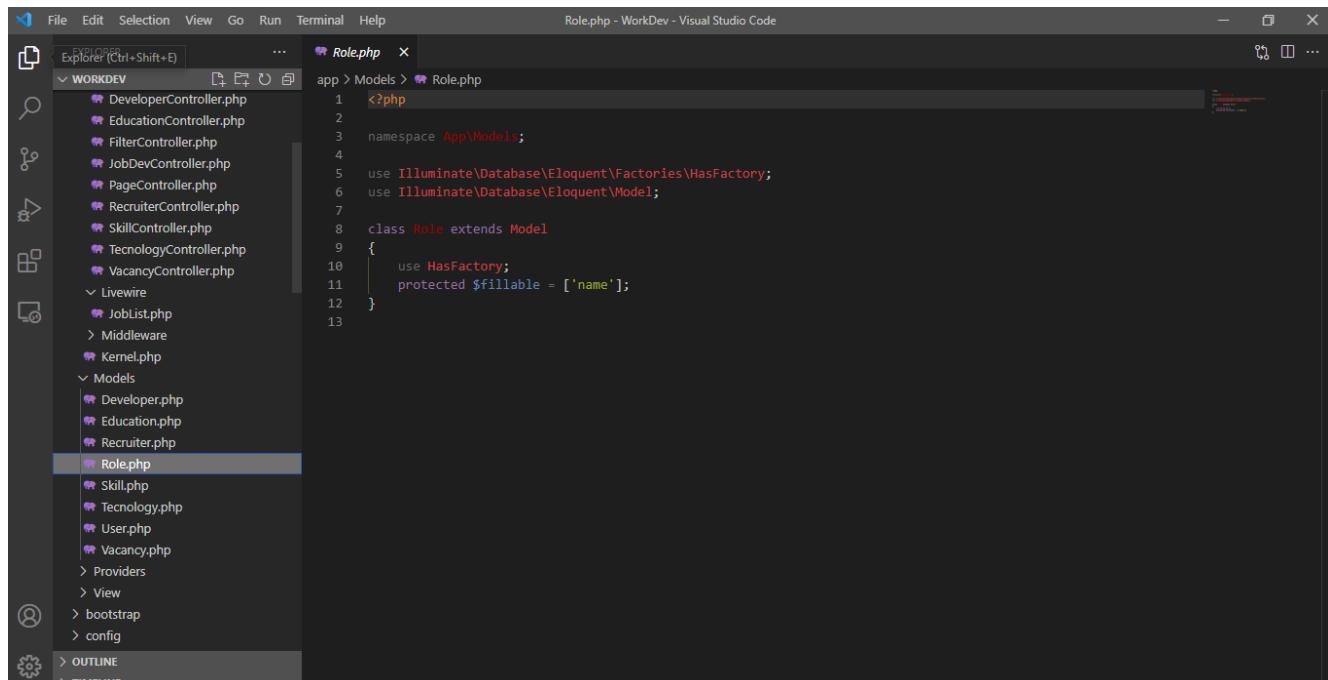
WORKEDEV

- DeveloperController.php
- EducationController.php
- FilterController.php
- JobDevController.php
- PageController.php
- RecruiterController.php
- SkillController.php
- TecnologyController.php
- VacancyController.php
- Livewire
- JobList.php
- Middleware
- Kernel.php
- Models
- Developer.php
- Education.php
- Recruiter.php
- Role.php
- Skill.php
- Tecnology.php
- User.php
- Vacancy.php
- Providers
- View
- bootstrap
- config

OUTLINE

TIMELINE

```
4 use Illuminate\Database\Eloquent\Factories\HasFactory;
5 use Illuminate\Database\Eloquent\Model;
6
7 class Recruiter extends Model
8 {
9     use HasFactory;
10
11     public function user()
12     {
13         return $this->belongsTo(User::class);
14     }
15
16     public function vacancies()
17     {
18         return $this->hasMany(Vacancy::class);
19     }
20 }
21
22 }
```



File Edit Selection View Go Run Terminal Help

Role.php - WORKDEV - Visual Studio Code

EXPLORER (Ctrl+Shift+E) ...

WORKEDEV

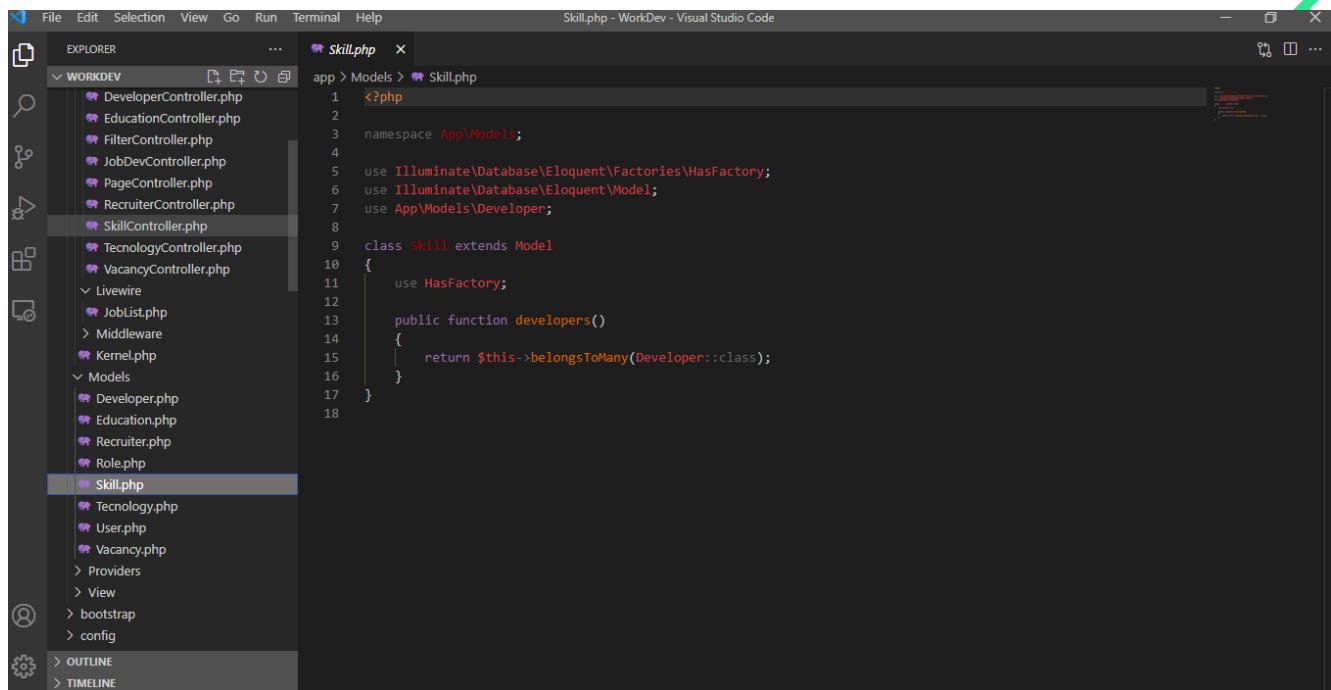
- DeveloperController.php
- EducationController.php
- FilterController.php
- JobDevController.php
- PageController.php
- RecruiterController.php
- SkillController.php
- TecnologyController.php
- VacancyController.php
- Livewire
- JobList.php
- Middleware
- Kernel.php
- Models
- Developer.php
- Education.php
- Recruiter.php
- Role.php
- Skill.php
- Tecnology.php
- User.php
- Vacancy.php
- Providers
- View
- bootstrap
- config

OUTLINE

TIMELINE

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Role extends Model
9 {
10     use HasFactory;
11     protected $fillable = ['name'];
12 }
```

Ejecución Modelos



File Edit Selection View Go Run Terminal Help

Skill.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV

```
app > Models > Skill.php
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use App\Models\Developer;
8
9 class Skill extends Model
10 {
11     use HasFactory;
12
13     public function developers()
14     {
15         return $this->belongsToMany(Developer::class);
16     }
17 }
18
```

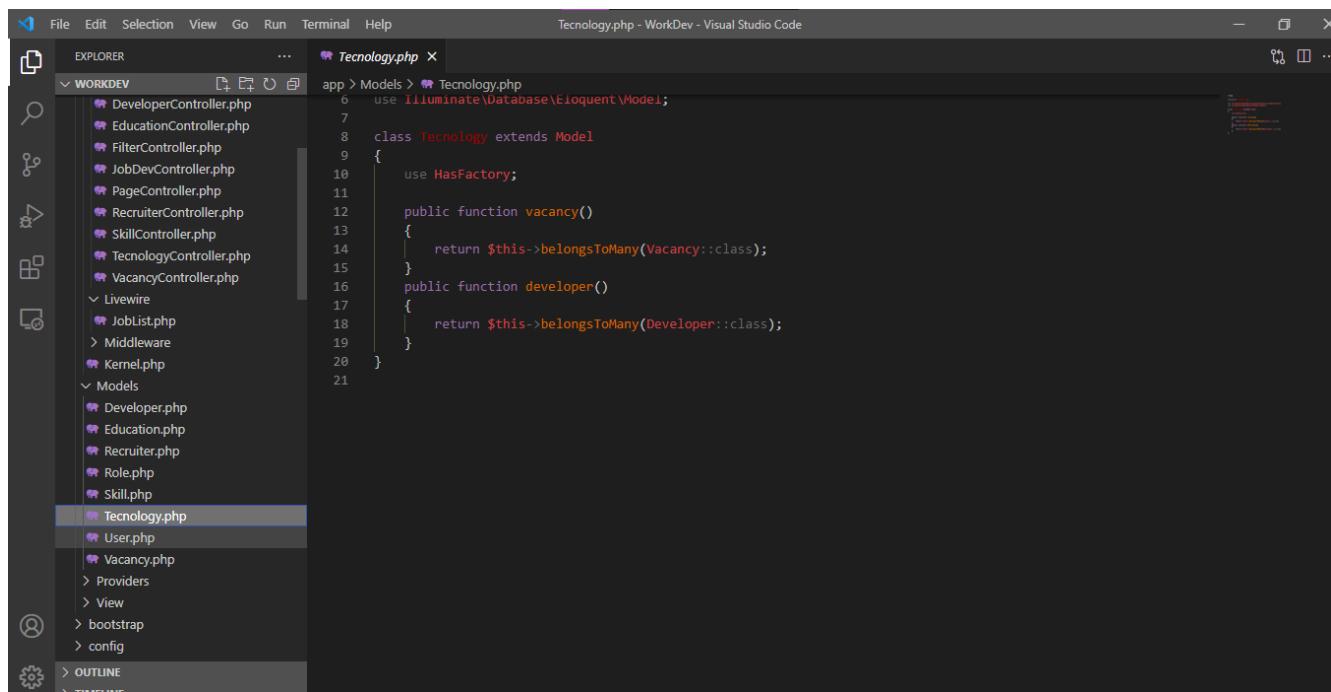
Livewire

Models

OUTLINE

TIMELINE

This screenshot shows the Visual Studio Code interface with the Skill.php file open. The file content defines a Skill model that belongs to many Developers. The code uses Eloquent factories and belongsToMany relationships.



File Edit Selection View Go Run Terminal Help

Tecnology.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV

```
app > Models > Tecnology.php
6 use Illuminate\Database\Eloquent\Model;
7
8 class Tecnology extends Model
9 {
10     use HasFactory;
11
12     public function vacancy()
13     {
14         return $this->belongsToMany(Vacancy::class);
15     }
16     public function developer()
17     {
18         return $this->belongsToMany(Developer::class);
19     }
20 }
```

Livewire

Models

OUTLINE

TIMELINE

This screenshot shows the Visual Studio Code interface with the Tecnology.php file open. The file content defines a Tecnology model that belongs to many Vacancies and Developers. The code uses Eloquent factories and belongsToMany relationships.

Ejecución Modelos

User.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER User.php x
WORKDEV app > Models > User.php
DeveloperController.php
EducationController.php
FilterController.php
JobDevController.php
PageController.php
RecruiterController.php
SkillController.php
TechnologyController.php
VacancyController.php
Livewire
JobList.php
> Middleware
Kernel.php
Models
Developer.php
Education.php
Recruiter.php
Role.php
Skill.php
Technology.php
User.php
Vacancy.php
> Providers
> View
> bootstrap
> config
> OUTLINE
> TIMELINE
    70     public function adminlte_desc(){
    71         return 'Reclutador/Empresa';
    72     }
    73
    74
    75
    76 }
    77

```

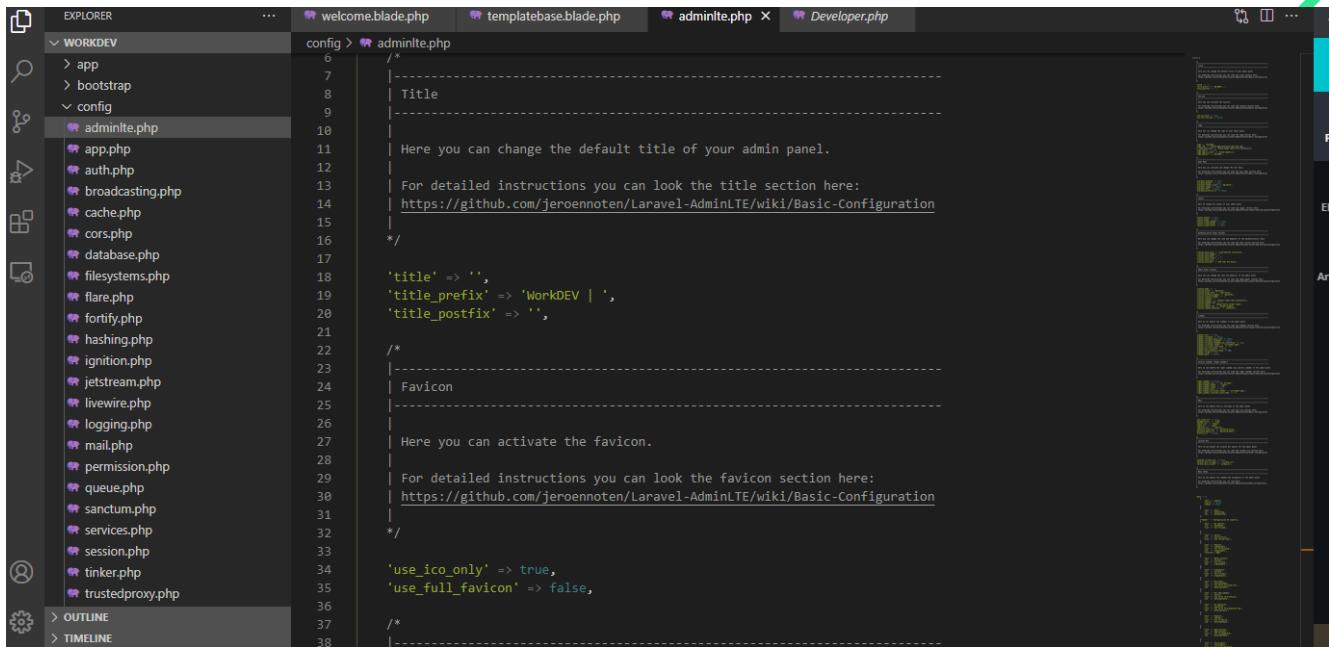
Vacancy.php - WorkDev - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
EXPLORER Vacancy.php x
WORKDEV app > Models > Vacancy.php
DeveloperController.php
EducationController.php
FilterController.php
JobDevController.php
PageController.php
RecruiterController.php
SkillController.php
TechnologyController.php
VacancyController.php
Livewire
JobList.php
> Middleware
Kernel.php
Models
Developer.php
Education.php
Recruiter.php
Role.php
Skill.php
Technology.php
User.php
Vacancy.php
> Providers
> View
> bootstrap
> config
> OUTLINE
> TIMELINE
    1 <?php
    2
    3 namespace App\Models;
    4
    5 use Illuminate\Database\Eloquent\Factories\HasFactory;
    6 use Illuminate\Database\Eloquent\Model;
    7
    8 class Vacancy extends Model
    9 {
    10     use HasFactory;
    11
    12     public function recrutar(){
    13         return $this->belongsTo(Recruiter::class);
    14     }
    15
    16     public function tecno(){
    17         return $this->hasMany(Technology::class);
    18     }
    19     public function developers(){
    20         return $this->belongsToMany(Developer::class);
    21     }
    22 }
    23
    24

```

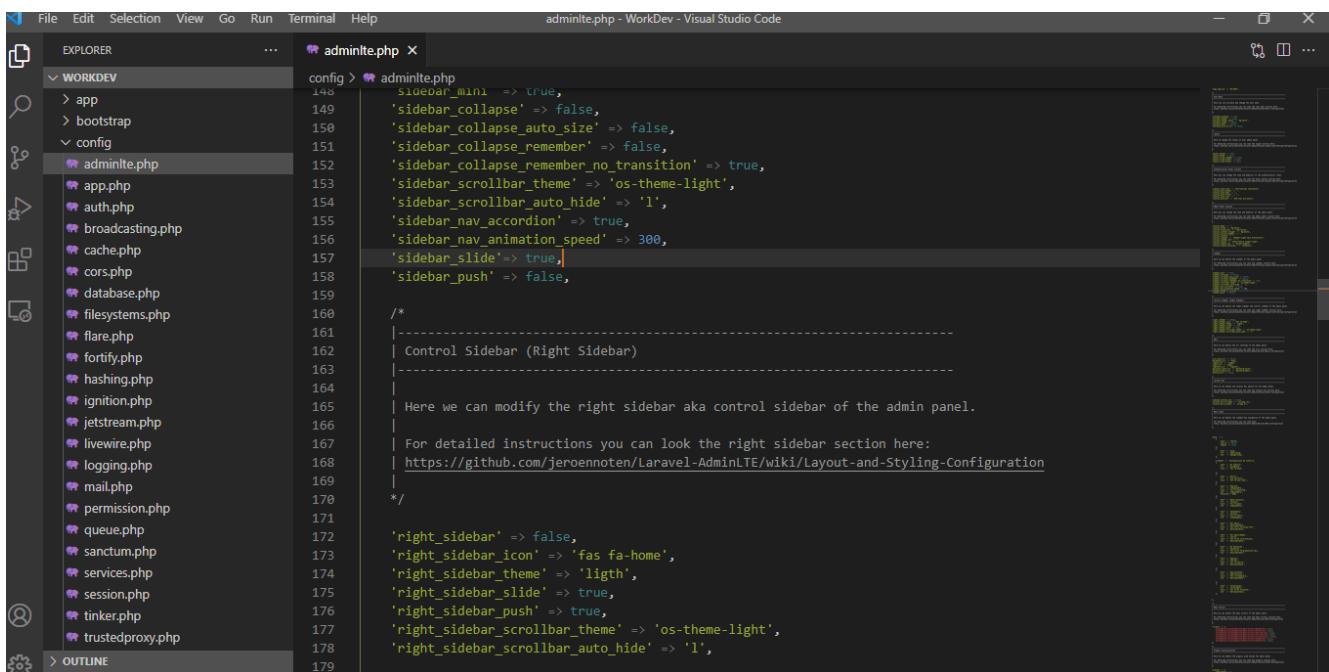
Ejecución AdminLTE



```

EXPLORER          ...   welcome.blade.php   templatebase.blade.php   adminlte.php x   Developer.php
WORKDEV
> app
> bootstrap
< config
  > adminlte.php
    app.php
    auth.php
    broadcasting.php
    cache.php
    cors.php
    database.php
    filesystems.php
    flare.php
    fortify.php
    hashing.php
    ignition.php
    jetstream.php
    livewire.php
    logging.php
    mail.php
    permission.php
    queue.php
    sanctum.php
    services.php
    session.php
    tinker.php
    trustedproxy.php
  > OUTLINE
  > TIMELINE
config > adminlte.php
  6   /*
  7   | -----
  8   | Title
  9   | -----
 10  | Here you can change the default title of your admin panel.
 11  |
 12  | For detailed instructions you can look the title section here:
 13  | https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Basic-Configuration
 14  |
 15  */
 16  */
 17  'title' => '',
 18  'title_prefix' => 'WorkDEV | ',
 19  'title_postfix' => '',
 20  /*
 21  | -----
 22  | Favicon
 23  | -----
 24  |
 25  | Here you can activate the favicon.
 26  |
 27  | For detailed instructions you can look the favicon section here:
 28  | https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Basic-Configuration
 29  |
 30  */
 31  /*
 32  */
 33  'use_ico_only' => true,
 34  'use_full_favicon' => false,
 35  /*
 36  */
 37  /*
 38  */

```

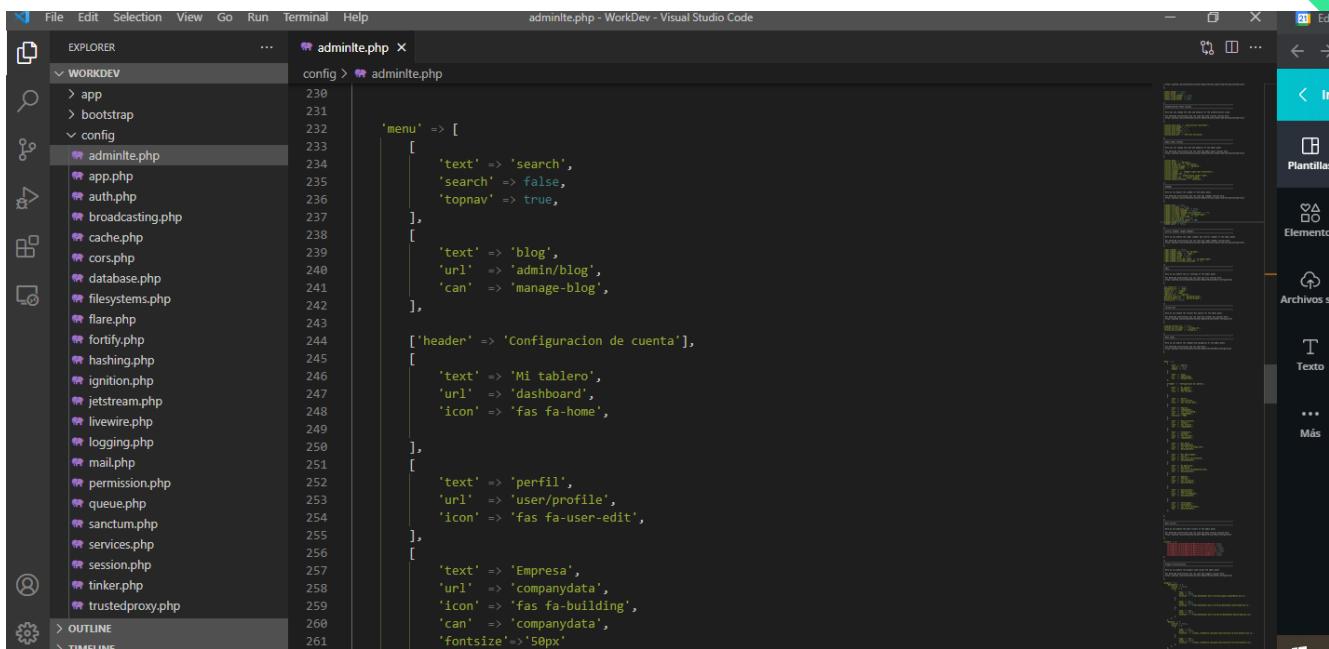


```

File Edit Selection View Go Run Terminal Help      adminlte.php - WorkDev - Visual Studio Code
EXPLORER          ...   adminlte.php x
WORKDEV
> app
> bootstrap
< config
  > adminlte.php
    app.php
    auth.php
    broadcasting.php
    cache.php
    cors.php
    database.php
    filesystems.php
    flare.php
    fortify.php
    hashing.php
    ignition.php
    jetstream.php
    livewire.php
    logging.php
    mail.php
    permission.php
    queue.php
    sanctum.php
    services.php
    session.php
    tinker.php
    trustedproxy.php
  > OUTLINE
  > TIMELINE
config > adminlte.php
  148 'sidebar_min' => true,
  149 'sidebarCollapse' => false,
  150 'sidebarCollapse_auto_size' => false,
  151 'sidebarCollapse_remember' => false,
  152 'sidebarCollapse_remember_no_transition' => true,
  153 'sidebar_scrollbar_theme' => 'os-theme-light',
  154 'sidebar_scrollbar_auto_hide' => '1',
  155 'sidebar_nav_accordion' => true,
  156 'sidebar_nav_animation_speed' => 300,
  157 'sidebar_slide' => true,
  158 'sidebar_push' => false,
  159 /*
 160 */
 161 /*
 162 | Control Sidebar (Right Sidebar)
 163 */
 164 /*
 165 | Here we can modify the right sidebar aka control sidebar of the admin panel.
 166 |
 167 | For detailed instructions you can look the right sidebar section here:
 168 | https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Layout-and-Styling-Configuration
 169 |
 170 */
 171 /*
 172 'right_sidebar' => false,
 173 'right_sidebar_icon' => 'fas fa-home',
 174 'right_sidebar_theme' => 'light',
 175 'right_sidebar_slide' => true,
 176 'right_sidebar_push' => true,
 177 'right_sidebar_scrollbar_theme' => 'os-theme-light',
 178 'right_sidebar_scrollbar_auto_hide' => '1',
 179 */

```

Ejecución AdminLTD



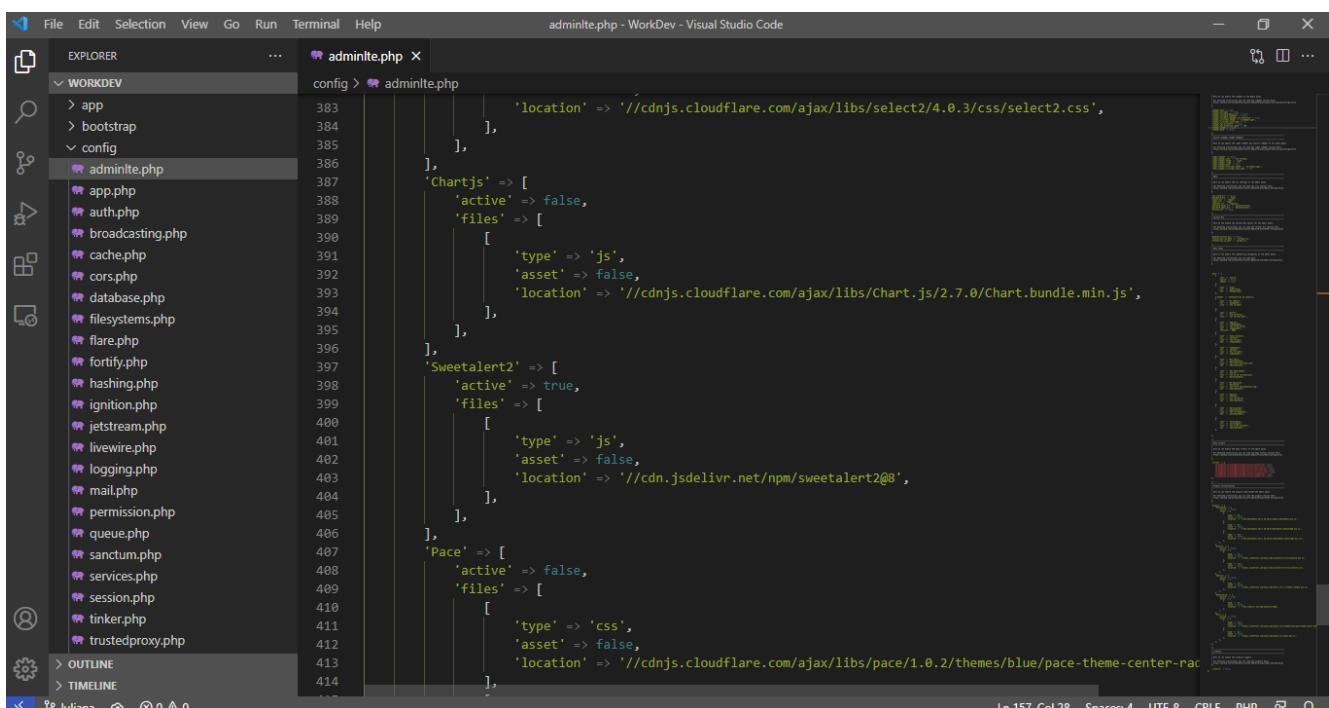
```

File Edit Selection View Go Run Terminal Help
adminlte.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV
config > adminlte.php
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261

'menu' => [
  [
    'text' => 'search',
    'search' => false,
    'topnav' => true,
  ],
  [
    'text' => 'blog',
    'url' => 'admin/blog',
    'can' => 'manage-blog',
  ],
  [
    'header' => 'Configuracion de cuenta'],
    [
      'text' => 'Mi tablero',
      'url' => 'dashboard',
      'icon' => 'fas fa-home',
    ],
    [
      'text' => 'perfil',
      'url' => 'user/profile',
      'icon' => 'fas fa-user-edit',
    ],
    [
      'text' => 'Empresa',
      'url' => 'companydata',
      'icon' => 'fas fa-building',
      'can' => 'companydata',
      'fontsize'=>'50px'
    ]
]

```



```

File Edit Selection View Go Run Terminal Help
adminlte.php - WorkDev - Visual Studio Code

EXPLORER WORKDEV
config > adminlte.php
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414

  'location' => '//cdnjs.cloudflare.com/ajax/libs/select2/4.0.3/css/select2.css',
  ],
],
],
'Chartjs' => [
  'active' => false,
  'files' => [
    [
      'type' => 'js',
      'asset' => false,
      'location' => '//cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.0/Chart.bundle.min.js',
    ],
  ],
],
'Sweetalert2' => [
  'active' => true,
  'files' => [
    [
      'type' => 'js',
      'asset' => false,
      'location' => '//cdn.jsdelivr.net/npm/sweetalert2@8',
    ],
  ],
],
'Pace' => [
  'active' => false,
  'files' => [
    [
      'type' => 'css',
      'asset' => false,
      'location' => '//cdnjs.cloudflare.com/ajax/libs/pace/1.0.2/themes/blue/pace-theme-center-rac

```

Ejecución rutas

File Edit Selection View Go Run Terminal Help

web.php

```

routes > web.php
35 Route::get('/', [PageController::class, 'home'])->name('home');
36
37 Route::resource('/ofertas', JobDevController::class);
38
39 Route::get('/ofert/filterBy', [FilterController::class, 'filterby'])->name('filterby');
40
41 Route::get('/nosotros', [PageController::class, 'nosotros'])->name('nosotros');
42
43 Route::resource('companydata', RecruiterController::class);
44
45 Route::get('/developerdata', [PageController::class, 'developerdata'])->name('developerdata');
46
47 Route::resource('skills', SkillController::class);
48
49 Route::resource('recruiter', RecruiterController::class);
50
51 Route::resource('developer', DeveloperController::class);
52
53 Route::resource('education', EducationController::class);
54
55 Route::resource('tecnologies', TecnologyController::class);
56
57 Route::resource('applications', ApplicationController::class);
58
59 //Route::resource('candidates', CandidateController::class);
60
61 Route::get('/board', [RecruiterController::class, 'board'])->name('board');
62
63 Route::get('/editDeveloper', [PageController::class, 'editDeveloper'])->name('editdeveloper');
64
65 Route::get('/candidates/{id:id}', [PageController::class, 'getCandidates'])->name('candidates');
66
67 Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {

```

File Edit Selection View Go Run Terminal Help

jetstream.php

```

routes > jetstream.php
9 use Laravel\Jetstream\Http\Controllers\Livewire\UserProfileController;
10 use Laravel\Jetstream\Http\Controllers\TeamInvitationController;
11 use Laravel\Jetstream\Jetstream;
12
13 Route::group(['middleware' => config('jetstream.middleware', ['web'])], function () {
14     if (Jetstream::hasTermsAndPrivacyPolicyFeature()) {
15         Route::get('/terms-of-service', [TermsOfServiceController::class, 'show'])->name('terms.show');
16         Route::get('/privacy-policy', [PrivacyPolicyController::class, 'show'])->name('policy.show');
17     }
18
19     Route::group(['middleware' => ['auth', 'verified']], function () {
20         // User & Profile...
21         Route::get('/user/profile', [UserProfileController::class, 'show'])
22             | ->name('profile.show');
23
24         // API...
25         if (Jetstream::hasApiFeatures()) {
26             Route::get('/user/api-tokens', [ApiClientController::class, 'index'])->name('api-tokens.index');
27         }
28
29         // Teams...
30         if (Jetstream::hasTeamFeatures()) {
31             Route::get('/teams/create', [TeamController::class, 'create'])->name('teams.create');
32             Route::get('/teams/{team}', [TeamController::class, 'show'])->name('teams.show');
33             Route::put('/current-team', [CurrentTeamController::class, 'update'])->name('current-team.update');
34
35             Route::get('/team-invitations/{invitation}', [TeamInvitationController::class, 'accept'])
36                 | ->middleware(['signed'])
37                 | ->name('team-invitations.accept');
38         }
39     });
40 });

```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF PHP ⌂ ⌂

Evaluación

Experiencia de usuario e interfaz

Persistencia de datos

Seguridad

Conclusión

Sustentación por competencias

Crear mock-ups para la aplicación

Front-end

El mock-up toma en cuenta las características funcionales descritas en los casos de uso o escenarios de usuario. El mock-up cumple con los principios de seguridad de interfaz de usuario. El contenido del mock-up está redactado en Español, de manera adecuada para el usuario y sin errores. Utiliza aplicaciones de diseño y/o maquetación profesionales y el mock-up toma en cuenta la experiencia de usuario y el flujo de la aplicación.

Para la realización del mock-up lo desarrolle en FIGMA

tuve en cuenta las vistas planificadas según el brief de proyecto las cuales son

- Página Principal
- Inicio de sesión
- Registro de un usuario
- Edición de perfil
- Vista de perfil de usuario
- Categoría vacantes
- Detalle de vacante
- Panel de control de vacantes
- Registro de una vacante
- Candidatos

por otra parte también tuve en cuenta una paleta de color predefinida que se mantiene en el desarrollo

A continuación comparto pdf del Mock-up

https://drive.google.com/file/d/1k9iSKhycS3z103NuBe_DG4pJvp8R-_ZF/view?usp=sharing

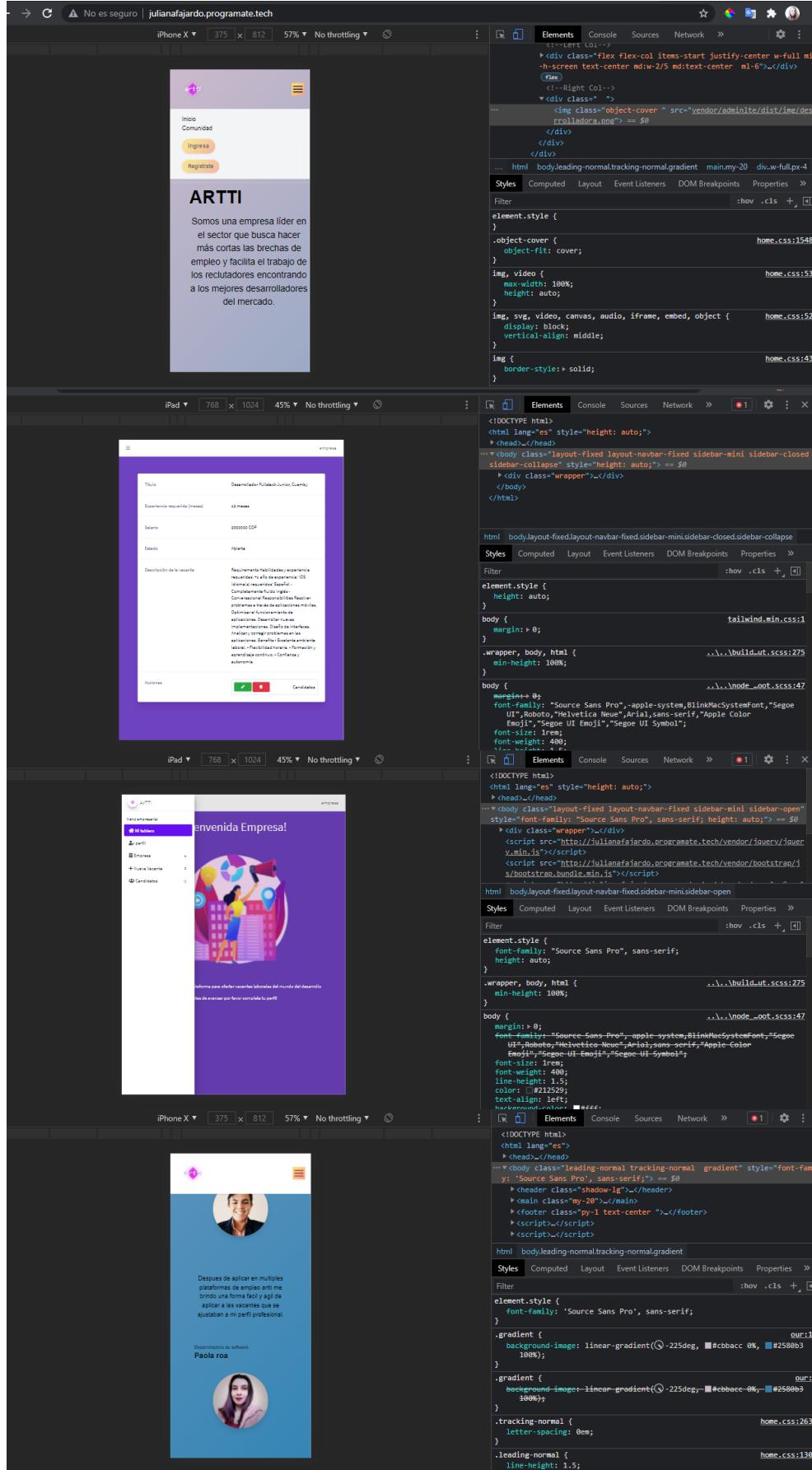
Sustentación

Front-end

Crear interfaces de usuario web estaticas y adaptativas

La interfaz se ajusta al mock-up diseñado. Se siguen las mejores prácticas estructurales, incluso para la web móvil. Las páginas web se adaptan al tamaño de la pantalla. El sitio web sigue las reglas de SEO técnico. El procedimiento de búsqueda les permite resolver un problema técnico o implementar una nueva característica. Se entiende la documentación técnica relativa a las tecnologías asociadas, redactada en español (sin espacio a interpretaciones erradas, etc.) Maneja media queries y distintos tipos de displays en css para lograr el objetivo.

En cuanto a la interfaz se ajusta en su versión móvil de diferentes plataformas las media queris vienen integradas con el tailwinds los cuales me facilitaron los estilos en cada una de las vistas Por ultimo la documentación esta presente en el presente documento y en algunas secciones del codigo.



iPhone X ▾ 375 x 812 57% No throttling

The screenshot shows a mobile browser interface with a sidebar menu open on the left. The main content area displays a form titled "Actualiza contraseña". The form instructions advise users to use a long and random password for security. It contains three input fields: "Contraseña actual", "Nueva contraseña", and "Confirmar contraseña", followed by a "GUARDAR" button. Below the form is a "Sesiones del navegador" section with a link to "Administrá y cierra la sesión de sus". The browser's developer tools are visible at the top, showing the DOM structure and CSS styles for the page.

iPhone X ▾ 375 x 812 57% No throttling

The screenshot shows a mobile browser interface displaying a landing page. The header features a logo and the text "Empresas que confian en nosotros". Below the header is a large graphic of a bar chart. A sidebar on the left lists "Desarrollador Fullstack Junior, Cuemby Gala" and provides localization details: "Localización: Colombia", "Salario: 2000000", and "Moneda: COP". The main content area includes a "Grad" section with a blue gradient background. The browser's developer tools are visible at the top, showing the DOM structure and CSS styles for the page.

→ C ▾ No es seguro | julianafajardo.programate.tech/login

iPhone X ▾ 375 x 812 57% No throttling

The screenshot shows a mobile browser interface displaying a login form. The header indicates "No es seguro". The form has fields for "Email" and "Contraseña", with a radio button for "Recuérdame". A "INICIAR SESIÓN" button is present. The background features a decorative illustration of a person with purple hair. The browser's developer tools are visible at the top, showing the DOM structure and CSS styles for the page.

Sustentación

Front-end

Desarrollar una interfaz de usuario web dinámica.

Las páginas web se ajustan a la experiencia del usuario, incluso para la experiencia móvil.

La arquitectura de la aplicación sigue las mejores prácticas de seguridad y desarrollo de aplicaciones web.

La aplicación web está optimizada para dispositivos móviles.

El procedimiento de búsqueda les permite resolver un problema técnico o implementar una nueva característica.

La vigilancia de las vulnerabilidades conocidas les permite identificar posibles fallas.

Se entiende la documentación técnica relativa a las tecnologías asociadas, redactada en español (sin espacio a interpretaciones erradas, etc.) Utiliza Javascript para implementar funcionalidades en el sitio web.

La interfaz es dinámica para el usuario ya que dependiendo el su rol cambia la interfaz de administración y cuenta con un login para verificar y validar al usuario y las rutas se encuentran protegidas dependiendo del rol y que este registrado

Sustentación

Front-end

Crear una interfaz de usuario usando un gestor

Las cuentas de usuario se crean con sus roles y permisos de acuerdo a su capa de accesibilidad.

La estructura del sitio web satisface las necesidades del cliente.

El sitio web sigue las reglas del SEO técnico.

Este gestor se da mediante cruds esencialmente mas sin embargo use Adminlte que es un paquete muy util de laravel 8 para que los usuarios cuenten con una sección de gestión, por otra parte tambien segun el rol en la base de datos se conecta a los controladores para que cada rol tenga un panel diferente

Sustentación

Back-end

Crear una base de datos

La base de datos se ajusta al esquema físico.

Los scripts de creación de base de datos e inserción de datos de prueba se ejecutan sin errores

Las necesidades de seguridad del DBMS se expresan de acuerdo con el estado de la técnica y los requisitos de seguridad identificados.

El procedimiento de búsqueda les permite resolver un problema técnico o implementar una nueva característica.

Se entiende la documentación técnica relativa a las tecnologías asociadas, redactada en español (sin malas interpretaciones, etc.) Manaja Sql para crear bases de datos y realizar queries.

La base de datos en su esquema inicial creció ya que se crearon tablas intermedias.

Mas sin embargo se ejecutan las debidas consultas y trae la información que conectan con las migraciones y conecta con los modelos

Conclusiones

las migraciones en Laravel se usan para generar de manera automática las diversas tablas en la base de datos, es decir a medida que se construye la aplicación se van creando las migración a su vez tablas en el base de datos este sistema permite tener un mejor sistema de control de versiones y en cuestiones de trabajo colaborativo permite tener diferentes interacciones y cambios sin necesidad de intervenir manualmente con la base de datos de otro usuario, ya que al crear las migraciones y subirlas en un repositorio las demás personas obtiene la información con solo actualizar y correr las migraciones.

Una vez que un trabajo haya superado esta cantidad de intentos, se inserta en la tabla de la base de datos failed_jobs.

Dentro de los modelos se generan las relaciones entre las tablas, como se hace, se llama la tabla que queremos conectar y dentro de este llamado se generan una o más funciones con otras tablas que asignen las relaciones proporcionadas por eloquent.

Por lo cual los controladores son archivos en los cuales se confirma el usuario con respecto a los datos integramos en la base en un momento inicial, con estos datos confirmados se procede a generar una o más consultas dentro de la misma que sean requeridas para diferentes permisos y funcionalidades.