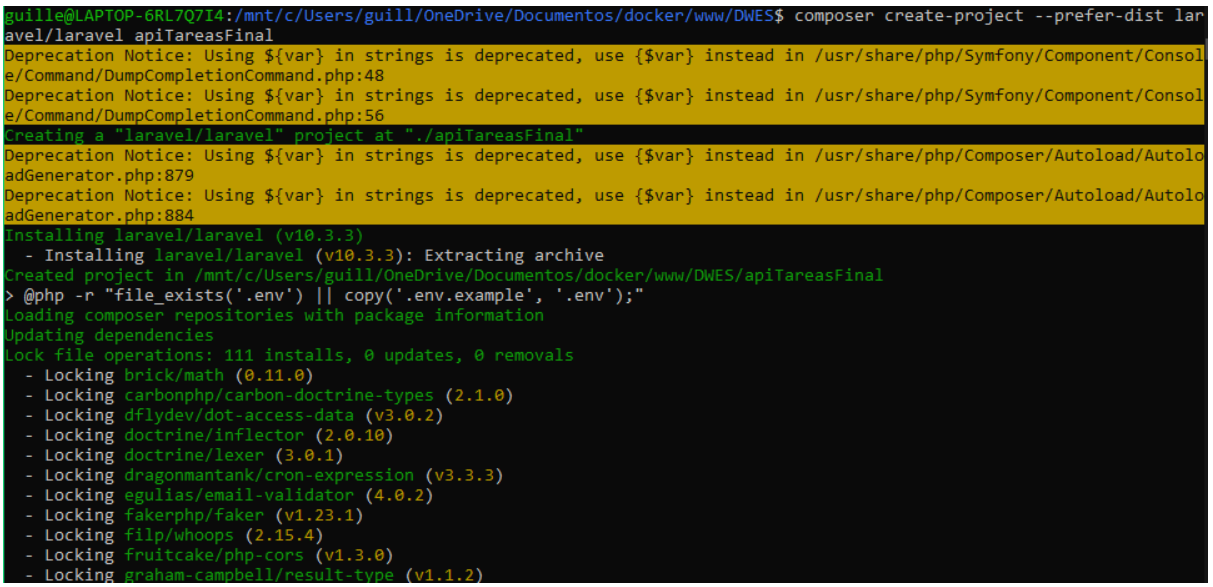
Para empezar a trabajar en Laravel necesitamos crear el proyecto en primera instancia. Para eso mediante cd nos introduciremos en la carpeta en la que queremos instalar el proyecto.

Una vez que estemos dentro de la carpeta en la que instalaremos el proyecto necesitaremos instalar composer, mediante el comando “composer install”, seguido de este comando necesitaremos poner el que nos instalará el proyecto el cual es “composer create-project --prefer-dist laravel/laravel apiTareasFinal” en este caso el nombre que se le ha dado es apiTareas, pero el nombre puede ser el que deseemos.

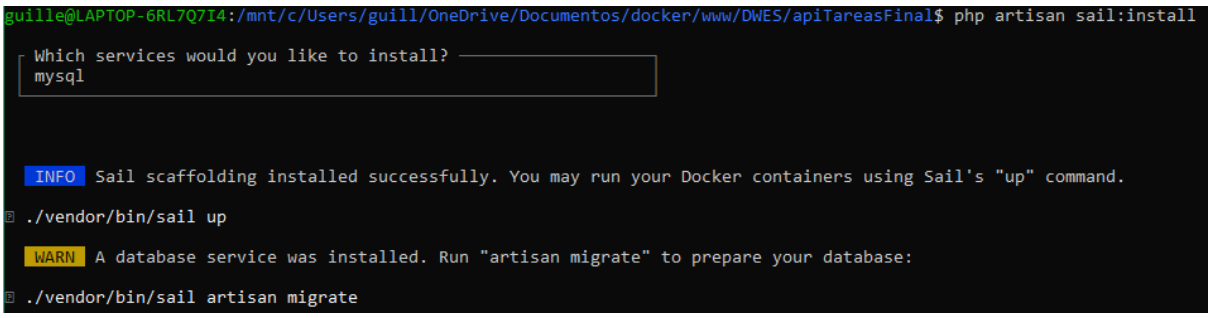
Al introducir este comando nos saldrá lo siguiente:



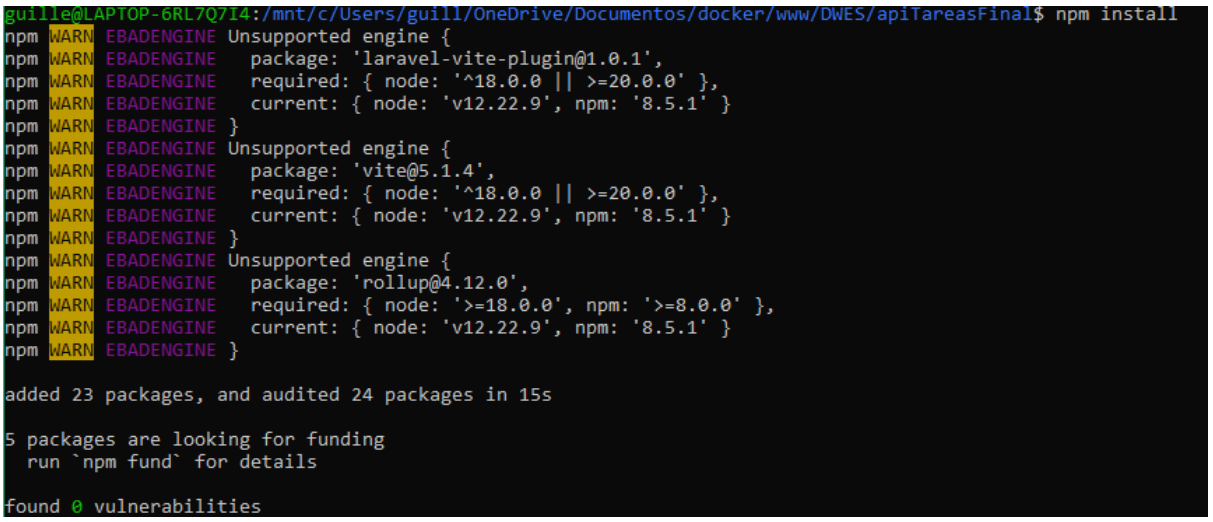
Cuando el proyecto esté instalado tendremos que introducirnos en la carpeta que se nos ha creado, la cual lleva el nombre del proyecto. En este caso será “cd apiTareasFinal”.

Una vez estemos dentro de esa carpeta deberemos poner el comando “php artisan sail:install”, el cual instala y configura Docker dentro del proyecto Laravel.

Cuando lo introducimos saldrá lo siguiente:



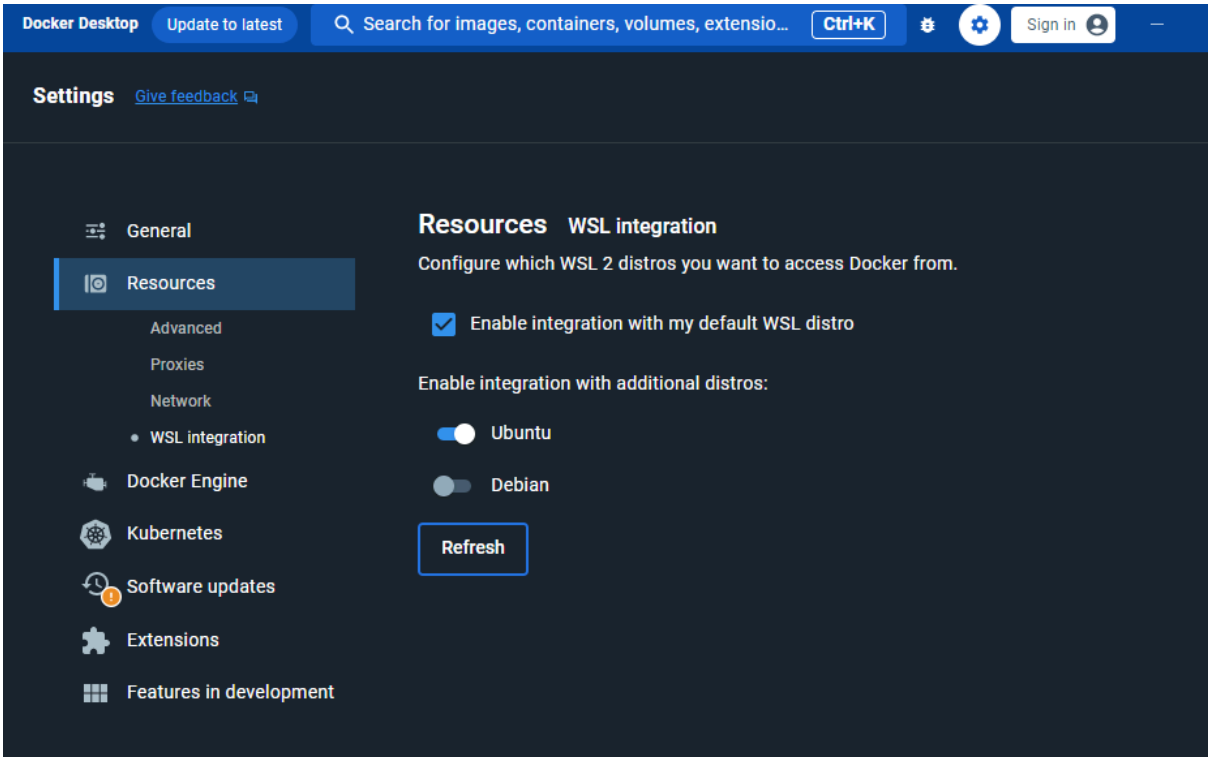
Una vez esté instalado correctamente deberemos poner “npm install” el cual instala todas las dependencias que le hacen falta al proyecto. Al ponerlo saldrá esto:



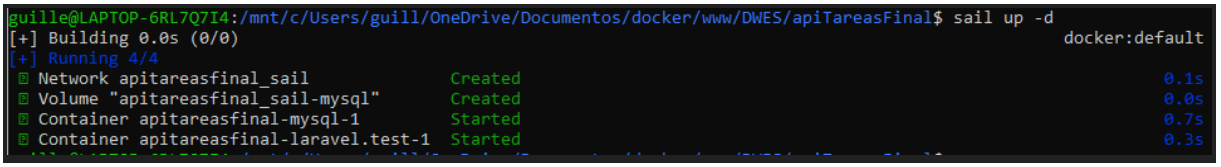
Después de instalar npm deberemos poner “alias sail=./vendor/bin/sail” lo cual hace que es crear un atajo con el nombre “sail” el cual hace referencia al ejecutable ./vendor/bin/sail, para así no tener que poner la ruta del ejecutable cada vez que queremos realizar una acción.

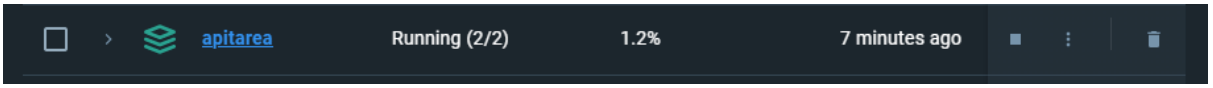
Una vez tengamos el alias definido deberemos poner el comando “sail up -d” el cual es un comando específico de Laravel que se utiliza para levantar los contenedores de Docker.

Pero antes de poner el comando tendremos que tener instalado en el ordenador el DockerDesktop el cual nos permite ver los contenedores de Docker que tenemos creados. Cuando lo tengamos instalado deberemos irnos a los ajustes y la parte de Resources y dentro de esta parte tendremos que acceder donde pone “WSL integration” y allí activar la pestaña de ubuntu ya que es desde donde lo estamos realizando.



Cuando nos aseguremos que esto está activado podremos poner el comando “sail up -d” lo que hará que en la terminal aparezca lo siguiente indicando que la creación del contenedor está realizada, además de poder verlo dentro del DockerDesktop.





Ahora que tenemos todo instalado correctamente, empezaremos a crear las tablas del proyecto.

Para empezar pondremos el comando “sail artisan make:migration create_tareas_table”, este comando realiza la migración para crear la tabla tareas en este caso ya que es el nombre que le hemos proporcionado.
Si la tabla se ha creado correctamente tendría que aparecer dentro de la carpeta /database/migrations con la fecha del día en que se esté realizando.

En este archivo que nos de le tendremos que decir los campos que queremos que tenga la tabla, en este caso le daremos id, titulo y descripción.

```php
public function up(): void
{
    Schema::create('tareas', function (Blueprint $table) {
        $table->id();
        $table->string("titulo", 40);
        $table->string("descripcion", 200)->nullable;
        $table->timestamps();
    });
}
```

Una vez esté creada la tabla tendremos que poner “sail artisan migrate” lo que hará que la tabla realice la migración al contenedor.

Cuando hayamos realizado este tendremos que realizar lo mismo pero con la tabla etiquetas, lo cual se realizara de la misma manera pero con la unica variacion de cambiar los nombres por etiquetas:

```
guille@LAPTOP-6RL7Q7I4:/mnt/c/Users/guill/OneDrive/Documentos/docker/www/DWES/apiTareasFinal$ sail artisan make:migratio
n create_etiquetas_table

  INFO  Migration [database/migrations/2024_03_03_211403_create_etiquetas_table.php] created successfully.
```

```php
    * Run the migrations.
    */
    public function up(): void
    {
        Schema::create('etiquetas', function (Blueprint $table) {
            $table->id();
            $table->string('nombre',20);
            $table->timestamps();
        });
    }

    /**
    * Reverse the migrations.
    */
    public function down(): void
    {
        Schema::dropIfExists('etiquetas');
    }
};
```

Ahora para finalizar las migraciones del proyecto tendremos que crear la tabla de “tareas_etiquetas”

```
guille@LAPTOP-6RL7Q7I4:/mnt/c/Users/guill/OneDrive/Documentos/docker/www/DWES/apiTareasFinal$ sail artisan make:migratio
n create_tareas_etiquetas_table

  INFO  Migration [database/migrations/2024_03_03_211851_create_tareas_etiquetas_table.php] created successfully.
```

```php
    public function up(): void
    {
        Schema::create('tareas_etiquetas', function (Blueprint $table) {
            $table->id();
            $table->foreignId('tareas_id')->constrained()->onDelete('cascade');
            $table->foreignId('etiquetas_id')->constrained()->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('tareas_etiquetas');
    }
};
```

```
guille@LAPTOP-6RL7Q7I4:/mnt/c/Users/guill/OneDrive/Documentos/docker/www/DWES/apiTareasFinal$ sail artisan migrate

  INFO  Running migrations.

  2024_03_03_211403_create_etiquetas_table ..................................................... 744ms DONE
  2024_03_03_211851_create_tareas_etiquetas_table ............................................. 2,551ms DONE
```

——————————————————————————————

Una ves que tengamos creados todas las migraciones del proyecto tendremos que crear los modelos, los cuales se utilizan para insertar información en las tablas.

Empezaremos por el modelo de Tarea el cual se crea ejecutando el comando “sail artisan make:model Tarea -cr”

```
guille@LAPTOP-6RL7Q7I4:/mnt/c/Users/guill/OneDrive/Documentos/docker/www/DWES/apiTareasFinal$ sail artisan make:model Ta
rea -cr

  INFO  Model [app/Models/Tarea.php] created successfully.

  INFO  Controller [app/Http/Controllers/TareaController.php] created successfully.
```

Despues de crear el modelo de Tarea crearemos el modelo de Etiqueta, lo cual se realizará de la misma manera que antes, poniendo el comando “sail artisan make:model Etiqueta -cr”

```
guille@LAPTOP-6RL7Q7I4:/mnt/c/Users/guill/OneDrive/Documentos/docker/www/DWES/apiTareasFinal$ sail artisan make:model Et
iqueta -cr

  INFO  Model [app/Models/Etiqueta.php] created successfully.

  INFO  Controller [app/Http/Controllers/EtiquetaController.php] created successfully.
```

Cuando hayamos creado los dos modelos nos habran aparecido los archivos de Tarea.php y Etiqueta.php dentro de la direccion “app/models” dentro de estos archivos tendremos que poner lo siguiente:

Tarea.php:

```php
class Tarea extends Model
{
    use HasFactory;

    0 references
    protected $fillable = ["nombre", "descripcion"];

    0 references
    protected $hidden = ["created_at", "updated_at"];

    0 references | 0 overrides
    public function etiquetas(): BelongsToMany{
        return $this->belongsToMany(Etiqueta::class, 'tareas_etiquetas', "tareas_id", "etiquetas_id");
    }
}
```

Etiqueta.php:

```php
class Etiqueta extends Model
{
    use HasFactory;

    0 references
    protected $fillable = ["nombre"];
    0 references
    protected $hidden = ["created_at", "updated_at"];

    0 references | 0 overrides
    public function tareas(): BelongsToMany{
        return $this->belongsToMany(Tarea::class, "tareas_etiqueta", "etiqueta_id", "tarea_id"  );
    }
}
```

Ahora crearemos los seeders los cuales realizan la inserción de los datos en la tabla.

Tendremos que crear tres seeder debido a que hemos creado tres tablas anteriormente.

El primer seeder que crearemos será el de Tarea, el cual se creará poniendo el comando “sail artisan make seeder TareasSeeder”

Una vez que el seeder esté creado nos encontraremos en la carpeta de “database/seeders” el seeder que acabamos de crear. Dentro de este seeder deberemos poner los datos que le queramos insertar a la tabla:

```php
public function run(): void
{
    DB::table('tareas')->insert([
        'titulo' => 'Hacer de comer',
        'descripcion' => 'Hacer macarrones con tomate'
    ]);

    DB::table('tareas')->insert([
        'titulo' => 'Recoger',
        'descripcion' => 'Recoger la colada'
    ]);

    DB::table('tareas')->insert([
        'titulo' => 'Entrenar',
        'descripcion' => 'Ir al gimnasio'
    ]);
}
```

Ahora crearemos el seeder de etiquetas que se realizará de la misma manera de las tareas pero cambiando los nombres mediante el comando “sail artisan make:seeder EtiquetasSeeder”

```php
public function run(): void
{
    DB::table('etiquetas')->insert([
        'nombre' => 'Cosas del hogar',
    ]);

    DB::table('etiquetas')->insert([
        'nombre' => 'Ayuda',
    ]);

    DB::table('etiquetas')->insert([
        'nombre' => 'Productividad',
    ]);
}
```

Tenemos realizado estos dos seeder ahora tendremos que crear el seeder de tarea_etiqueta. Este archivo será distinto ya que aquí tendremos que realizar la conexion de las otras dos tablas, para realizar esto se hará de la manera siguiente:
Primero tendremos que crear el seeder de la misma manera que hemos creado los otros dos “sail artisan make:seeder TareaEtiquetaSeeder”

Una vez tengamos creado el seeder tendremos que rellenar el archivo de la siguiente manera:

```php
public function run(): void
{
    $tareasComida= Tarea::where('titulo','Hacer de comer')->first();
    $tareasRecoger= Tarea::where('titulo', 'Recoger')->first();
    $tareasEntrenar= Tarea::where('titulo', 'Entrenar')->first();

    $etiquetasHogar= Etiqueta::where('nombre', 'Cosas del hogar')->first();
    $etiquetasAyuda= Etiqueta::where('nombre', 'Ayuda')->first();
    $etiquetasProductividad= Etiqueta::where('nombre', 'Productividad')->first();

    $tareasComida->etiquetas()->attach($etiquetasHogar);
    $tareasRecoger->etiquetas()->attach($etiquetasAyuda);
    $tareasEntrenar->etiquetas()->attach($etiquetasProductividad);
}
```

Con esto creamos varias variables en las que le damos los nombres de los datos que están en las tablas y los atamos a las otras tablas que decimos que hagan referencia.

Ahora tendremos que subir todos los cambios a las base, para eso dentro de DatabaseSeeder.php tendremos que poner lo siguiente:

```php
class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    0 references | 0 overrides
    public function run(): void
    {
        $this->call(TareasSeeder::class);
        $this->call(EtiquetasSeeder::class);
        $this->call(TareaEtiquetaSeeder::class);
    }
```

Una vez pongamos eso tendremos que ejecutar el comando “sail artisan db:seed”

Seguidamente tendremos que crear los seeder que son los que utilizaremos para rellenar la tabla, esto se crea con el comando “sail artisan make:seeder TareasSeeder”.

Una vez se ejecute este comando encontraremos dentro de /database/seeders un nuevo archivo el cual se llamará TareasSeeder.php el cual tendremos que rellenar con los datos que queramos.

Cuando la tabla esté rellenada tendremos que poner el comando “sail artisan db:seed“ lo que hará que las tablas se rellenen en la base de datos.

Una vez este comando este realizado, poniendo “sail mysql” y dentro de la interfaz que se nos abra ponemos visualizar que la inserción de los datos se ha realizado en la tabla poniendo “sail mysql”, a lo que nos saldrá algo así:

```
mysql> select * from tareas;
+----+----------------+-----------------------------+------------+------------+
| id | titulo         | descripcion                 | created_at | updated_at |
+----+----------------+-----------------------------+------------+------------+
|  1 | Hacer de comer | Hacer macarrones con tomate | NULL       | NULL       |
|  2 | Recoger        | Recoger la colada           | NULL       | NULL       |
|  3 | Entrenar       | Ir al gimnasio              | NULL       | NULL       |
```

//----------

AÑADIR ESTO