

Archivos

Ubicación de los componentes de Blade: `app/View/Components` Son componentes de Blade documentados en el URL: <https://laravel.com/docs/9.x/blade#components>

- La plantilla del formulario dinámico de registro:
`resources/views/formulario/index.blade.php`
- La plantilla del formulario dinámico de edición:
`resources/views/formulario/edicion.blade.php`

Las rutas (endpoints) de las funcionalidades del formulario "dinámico" están en el archivo (`routes/admin.php`) La siguiente ruta es la que muestra el formulario dinámico.

```
Route::get('/formulario/index/{id_cuestionario}', [FormularioController::class, 'index'])->name('formulario.index')->middleware('signed');
```

El controlador asociado es: `app/Http/Controllers/Admin/FormularioController.php` y la acción es el `index`

La acción del formulario para guardar los datos de un nuevo registro es:

```
Route::post('/respuestas', [FormularioController::class, 'guardaRespuestas'])->name('formulario.respuestas');
```

- Esta ruta recibe los datos del formulario por AJAX
- Utiliza el método `validaRespuestas($request, $preguntas);` para crear un validador de Laravel que validará las entradas. Este método está en el mismo Controlador `FormularioController`

Validación

- La obligatoriedad de una pregunta se guarda en la tabla `PREGUNTA` en el campo
- Se tendría que agregar el caso al switch en el método de `validaRespuesta()`

Guardado

- El guardado se realiza en el método `guardaRespuestas()` pero solo para el registro.
- El formulario de un nuevo registro: `resources/views/formulario/index.blade.php`
- El formulario de edición es: `resources/views/formulario/edicion.blade.php`

Los endpoints para la edición de un registro:

```
Route::get('/formulario/{id}/edicion', [FormularioController::class, 'edicion'])->>name('formulario.edicion');
Route::put('/editarrespuestas/{id}/', [FormularioController::class, 'editarrespuestas'])->>name('formulario.editarrespuestas');
```

Seeders

- Los seeders son para probar lo que desarrolles. Están en `database/seeders`
- Como no existe la interfaz para crear cuestionarios, se usan los seeders. En los seeders se define la estructura de los cuestionarios.

Para ejecutar un seeder:

```
php artisan db:seed --class=ColorSeeder
```

En este caso ColorSeeder es el nombre del seeder, están en el directorio `database/seeders`

Cuentas de usuario

Usuario

admin@noexiste.unam.mx

hola123.

ldanielbg@comunidad.unam.mx

hola123.

Archivo de constantes

- El archivo de constantes: `config/constantes.php`
- Las constantes son para validar qué tipo de pregunta es.

Flyway

- Son scripts SQL
- Están en el directorio: `database/flyway/sql`

Asociar un cuestionario a un usuario

- Para que un usuario pueda contestar un cuestionario es necesario agregarlo desde el sistema usando la cuenta de administrador.