

Informe de Laboratorio 10

Tema: Proyecto Final

Nota

Estudiante(s)	Escuela	Asignatura
Christian Raul Mestas Zegarra cmestasz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: I Código: 1702122

Laboratorio	Tema	Duración
10	Proyecto Final	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 24 Junio 2024	Al 06 Julio 2024

Laboratorio 10

1 Tarea	3
2 Equipos y materiales utilizados	4
3 Desarrollo	5
3.1 Backend en Django	5
3.2 Frontend en Angular	15
3.2.1 Aplicación base	15
3.2.2 dynamic-form	16
3.2.3 home	19
3.2.4 login	29
3.2.5 register	30
3.2.6 services	32
3.2.7 models	33
3.3 Demostración	34
3.4 Lista de commits	35
4 URL del repositorio en GitHub	38
5 Enlace al video	38
6 Estructura de laboratorio 10	38
7 Referencias	39

1 Tarea

- **Enunciado:**

- Este trabajo debe ser hecho en grupos de 3 a 4. Escoger una empresa, escoger el tema a trabajar y definir los Requerimientos funcionales.
- El trabajo en equipo, debe ser hecho de modo que todos los participantes del grupo conozcan todo el código.
 - * La programación debe ser hecha en pares, es decir que durante una videollamada, dos o más integrantes (al menos dos, pueden ser todos los integrantes del grupo) programen un mismo código en un repositorio central, turnándose la labor de observador y codificar a intervalos de tiempo equitativos.
 - * No se debe dividir el trabajo en partes y que los integrantes hagan su parte sin que los demás conozcan de su trabajo o él conozca y participe en el trabajo de los demás.
 - * Se deben crear ramas en sus proyectos de git para cada una de las funcionalidades, una vez alcanzada con éxito la funcionalidad, esta se debe integrar a la rama principal.
- La aplicación será calificada sobre 20 y deberá incluir:
 - * URLs propios, usando reverse (2 puntos)
 - * Plantillas propias de la aplicación (1 puntos)
 - * Que usen widgets de manera elegante (1 puntos)
 - * Vistas de Listado, Detalle, Crear, Actualizar y Borrar (4 puntos)
 - * Formulario con restricciones de seguridad adicionales (2 puntos)
 - * Vista de consultas que devuelva Json (3 puntos)
 - * Programa cliente para hacer y consumir las consultas con Ajax (2 puntos)
 - * Programa cliente para hacer y consumir las consultas con Framework de JavaScript (3 puntos)
 - * Al menos dos modelos (2 puntos)
 - * Modelo con clave externa: foreign key (+2 puntos, opcional)
 - * CSS o Bootstrap (2 puntos)
 - * Publicó su aplicación en el web (+3 puntos, opcional)
 - * Descargar un informe como archivos pdf (+2 puntos, opcional)
 - * Enviar correo (+2 puntos, opcional)
- Trabajo en equipo, los 20 puntos serán distribuidos de manera equitativa entre los siguientes criterios
 - * Responsabilidad: cumplió puntualmente con todo
 - * Proactividad: hizo más de lo que se pidió
 - * Aporte al grupo
 - * Calificó a sus compañeros (Si no lo hizo 50% de la nota de trabajo individual)
- La calificación final será un promedio entre la nota individual 40% y grupal 60%
 - * La nota grupal será por el producto calificado por el profesor, en base a sus avances, exposición e informe.
 - * La nota individual será calificada por sus propios compañeros. Los que no califiquen a sus compañeros tendrán la mitad de la nota en este componente.

2 Equipos y materiales utilizados

- Cuenta en GitHub con el correo institucional.
- Sistema Operativo Microsoft Windows 10
- Visual Studio Code
- Git
- Windows PowerShell
- Navegador Mozilla Firefox
- Angular CLI
- Django

3 Desarrollo

3.1 Backend en Django

models.py

```
1  from django.db import models
2  from django.contrib.auth.models import AbstractUser, BaseUserManager
3
4  # Create your models here.
5
6
7  class UsuarioManager(BaseUserManager):
8      def create_user(self, username, telefono, password):
9          if not username:
10             raise ValueError('Ingrese un correo electrónico')
11          if not telefono:
12             raise ValueError('Ingrese un número de teléfono')
13          if not password:
14             raise ValueError('Ingrese una contraseña')
15          usuario = self.model(
16              username=self.normalize_email(username),
17              telefono=telefono
18          )
19          usuario.set_password(password)
20          usuario.save(using=self._db)
21          return usuario
22
23      def create_superuser(self, username, email, password=None):
24          usuario = self.create_user(username, '000000000', password)
25          usuario.is_admin = True
26          usuario.is_staff = True
27          usuario.is_superuser = True
28          usuario.save(using=self._db)
29          return usuario
30
31
32  class Usuario(AbstractUser):
33      class Types(models.TextChoices):
34          USUARIO = 'US', 'Usuario'
35          DUEÑO = 'DU', 'Dueño'
36          ADMIN = 'AD', 'Administrador'
37
38
39      username = models.EmailField(unique=True)
40      telefono = models.CharField(max_length=9)
41      yape_qr = models.ImageField(upload_to='yape_qrs/', blank=True, null=True)
42      tipo = models.CharField(max_length=2, choices=Types.choices, default=Types.USUARIO)
43
44      objects = UsuarioManager()
45
46      USERNAME_FIELD = 'username'
47
48
49  class Tienda(models.Model):
50      nombre = models.CharField(max_length=100)
51      descripcion = models.TextField()
```

```
52     categoria = models.CharField(max_length=100)
53     dueño = models.ForeignKey(Usuario, on_delete=models.CASCADE, limit_choices_to={'tipo':
54     ↪ Usuario.Types.DUEÑO})
54     latitud = models.DecimalField(max_digits=9, decimal_places=6)
55     longitud = models.DecimalField(max_digits=9, decimal_places=6)
56
57
58 class Producto(models.Model):
59     nombre = models.CharField(max_length=100)
60     descripcion = models.TextField()
61     precio = models.DecimalField(max_digits=6, decimal_places=2)
62     imagen = models.ImageField(upload_to='productos/', blank=True, null=True)
63     tienda = models.ForeignKey(Tienda, on_delete=models.CASCADE)
64     stock = models.PositiveIntegerField()
65
66
67 class Venta(models.Model):
68     usuario = models.ForeignKey(Usuario, on_delete=models.CASCADE, limit_choices_to={'tipo':
69     ↪ Usuario.Types.USUARIO})
69     producto = models.ForeignKey(Producto, on_delete=models.CASCADE)
70     fecha = models.DateTimeField(auto_now_add=True)
71     cantidad = models.PositiveIntegerField()
```

- Se crean los modelos **Usuario**, **Tienda**, **Producto** y **Venta**.
- Se crea la clase **UsuarioManager** para poder usar la autenticación de Django.

forms.py

```
1  from django import forms
2  from django.forms import ModelForm
3  from .models import Usuario, Tienda, Producto, Venta
4
5  class LoginForm(forms.Form):
6      username = forms.CharField(label='Nombre de usuario')
7      password = forms.CharField(label='Contraseña', widget=forms.PasswordInput)
8
9  class LogoutForm(forms.Form):
10     pass
11
12 class UsuarioForm(ModelForm):
13     password1 = forms.CharField(label='Contraseña', widget=forms.PasswordInput)
14     password2 = forms.CharField(label='Confirmar contraseña', widget=forms.PasswordInput)
15
16     class Meta:
17         model = Usuario
18         fields = ['username', 'telefono', 'password1', 'password2']
19
20     def clean_telefono(self):
21         telefono = self.cleaned_data.get('telefono')
22         if not telefono:
23             raise forms.ValidationError('Ingrese un número de teléfono')
24         if len(telefono) != 9:
25             raise forms.ValidationError('Ingrese un número de teléfono válido')
26
27     def clean_password2(self):
```

```
28         password1 = self.cleaned_data.get('password1')
29         password2 = self.cleaned_data.get('password2')
30         if password1 and password2 and password1 != password2:
31             raise forms.ValidationError('Las contraseñas no coinciden')
32         return password2
33
34     def save(self, commit=True):
35         user = super().save(commit=False)
36         user.set_password(self.cleaned_data['password1'])
37         if commit:
38             user.save()
39         return user
40
41 class DueñoForm(UsuarioForm):
42     class Meta:
43         model = Usuario
44         fields = ['username', 'telefono', 'password1', 'password2', 'yape_qr']
45
46     def save(self, commit=True):
47         user = super().save(commit=False)
48         user.tipo = Usuario.Types.DUEÑO
49         if commit:
50             user.save()
51         return user
52
53     def clean_yape_qr(self):
54         yape_qr = self.cleaned_data.get('yape_qr')
55         if not yape_qr:
56             instance = self.instance
57             if instance.pk:
58                 yape_qr = instance.yape_qr
59         return yape_qr
60
61 class AdminForm(UsuarioForm):
62     class Meta:
63         model = Usuario
64         fields = ['username', 'telefono', 'password1', 'password2']
65
66     def save(self, commit=True):
67         user = super().save(commit=False)
68         user.tipo = Usuario.Types.ADMIN
69         if commit:
70             user.save()
71         return user
72
73 class TiendaForm(ModelForm):
74     class Meta:
75         model = Tienda
76         fields = '__all__'
77
78 class ProductoForm(ModelForm):
79     class Meta:
80         model = Producto
81         fields = '__all__'
82
83
```

```
84 class VentaForm(ModelForm):
85     class Meta:
86         model = Venta
87         fields = '__all__'
```

- Se crean los formularios **UsuarioForm**, **DueñoForm**, **AdminForm**, **TiendaForm**, **ProductoForm** y **VentaForm**.
- Además se crean los formularios **LoginForm** y **LogoutForm** para el inicio y cierre de sesión.
- Los formularios tienen validaciones personalizadas para los campos.

serializers.py

```
1  from django import forms
2  from django.forms import ModelForm
3  from django.core import validators
4  from django.core.serializers.json import DjangoJSONEncoder
5  from django.db.models import Model
6  from django.db.models.fields.files import ImageFieldFile
7  from decimal import Decimal
8
9
10 class CustomJSONEncoder(DjangoJSONEncoder):
11     def default(self, obj):
12         if isinstance(obj, ImageFieldFile):
13             return obj.url if obj else None
14         if isinstance(obj, Decimal):
15             return float(obj)
16         return super().default(obj)
17
18
19 def model_serializer(instance: Model, fields=None, exclude=None):
20     data = {}
21     for field in instance._meta.fields:
22         if (fields is None or field.name in fields) and (exclude is None or field.name not in
23             ↪ exclude):
24             value = getattr(instance, field.name)
25             if isinstance(value, Model):
26                 value = model_serializer(value)
27             elif isinstance(value, ImageFieldFile):
28                 value = value.url if value else None
29             elif isinstance(value, Decimal):
30                 value = float(value)
31             data[field.name] = value
32     return data
33
34 def form_serializer(form: ModelForm):
35     fields = []
36     for name, field in form.fields.items():
37         field_data = {
38             'tipoCampo': get_tipo_campo(field),
39             'name': name,
40             'label': field.label or name.capitalize(),
41             'validators': get_validators(field),
```



```
42         'attributes': get_attributes(field),
43     }
44     if field_data['tipoCampo'] == 'select':
45         field_data['options'] = get_options(field)
46     fields.append(field_data)
47     return fields
48
49
50 def get_tipo_campo(field):
51     if isinstance(field, forms.FileField) or isinstance(field, forms.ImageField):
52         return 'input'
53     if isinstance(field, forms.ChoiceField):
54         return 'select'
55     if isinstance(field, forms.Textarea):
56         return 'textarea'
57     return 'input'
58
59
60 def get_validators(field):
61     validators_dict = {}
62     for validator in field.validators:
63         if isinstance(validator, validators.MaxLengthValidator):
64             validators_dict['maxlength'] = str(validator.limit_value)
65         elif isinstance(validator, validators.MinLengthValidator):
66             validators_dict['minlength'] = str(validator.limit_value)
67         elif isinstance(validator, validators.EmailValidator):
68             validators_dict['email'] = 'true'
69
70     if field.required:
71         validators_dict['required'] = 'true'
72
73     if isinstance(field, forms.DecimalField):
74         validators_dict['min'] = str(
75             field.min_value) if field.min_value is not None else ''
76         validators_dict['max'] = str(
77             field.max_value) if field.max_value is not None else ''
78         validators_dict['decimal'] = 'true'
79
80     if isinstance(field, forms.IntegerField):
81         validators_dict['min'] = str(
82             field.min_value) if field.min_value is not None else ''
83         validators_dict['max'] = str(
84             field.max_value) if field.max_value is not None else ''
85
86     return validators_dict
87
88
89 def get_attributes(field):
90     attributes = {}
91     if hasattr(field.widget, 'input_type'):
92         attributes['type'] = field.widget.input_type
93
94     if isinstance(field, forms.FileField) or isinstance(field, forms.ImageField):
95         attributes['type'] = 'file'
96     if isinstance(field, forms.ImageField):
97         attributes['accept'] = 'image/*'
```

```
98
99     if isinstance(field, forms.DateTimeField):
100         attributes['type'] = 'datetime-local'
101
102     if isinstance(field, forms.DecimalField):
103         attributes['step'] = str(10 ** -field.decimal_places)
104
105     return attributes
106
107
108 def get_options(field):
109     if isinstance(field, forms.ModelChoiceField):
110         return [{'label': str(obj), 'value': obj.pk} for obj in field.queryset]
111     elif hasattr(field, 'choices'):
112         return [{'label': label, 'value': serialize_value(value)} for value, label in
113                 ↪ field.choices]
114
115     return []
116
117 def serialize_value(value):
118     if hasattr(value, 'pk'):
119         return value.pk
120     elif isinstance(value, (list, tuple)) and len(value) == 2:
121         return value[0]
122     return value
```

- Se crea el serializador para formularios, que puede convertir todos los campos del formulario en un JSON.
- Se crea el serializador para los modelos, que puede convertir todos los campos del modelo en un JSON.

decorators.py

```
1 from django.http import JsonResponse
2 from functools import wraps
3
4 def api_login_required(func):
5     @wraps(func)
6     def wrapper(request, *args, **kwargs):
7         if request.user.is_authenticated:
8             return func(request, *args, **kwargs)
9         return JsonResponse({'status': 401, 'message': 'No permitido'})
10     return wrapper
```

- Se crea el decorador `api_login_required` para verificar si el usuario está autenticado al hacer las peticiones por api.

views.py

```
1 from django.shortcuts import render
2 from django.urls import reverse
3 from django.http import JsonResponse
4 from django.contrib.auth.decorators import login_required
5 from .forms import UsuarioForm, DueñoForm, TiendaForm, ProductoForm, VentaForm, LoginForm,
↪ LogoutForm
```

```
6 from .models import Tienda, Producto, Venta, Usuario
7 from .decorators import api_login_required
8 from .serializers import form_serializer, model_serializer
9 from django.contrib.auth import authenticate, login, logout
10
11 # Create your views here.
12
13
14 def forms_test(request):
15     context = {
16         'forms': [
17             {'form': LoginForm(), 'url': reverse('iniciar_sesion')},
18             {'form': LogoutForm(), 'url': reverse('cerrar_sesion')},
19             {'form': UsuarioForm(), 'url': reverse('create_usuario')},
20             {'form': DueñoForm(), 'url': reverse('create_dueño')},
21             {'form': TiendaForm(), 'url': reverse('create_tienda')},
22             {'form': ProductoForm(), 'url': reverse('create_producto')},
23             {'form': VentaForm(), 'url': reverse('create_venta')},
24         ]
25     }
26     return render(request, 'forms_test.html', context)
27
28
29 def iniciar_sesion(request):
30     if request.method == 'POST':
31         form = LoginForm(request.POST)
32         if form.is_valid():
33             user = authenticate(
34                 request, username=form.cleaned_data['username'],
35                 password=form.cleaned_data['password'])
36             if user:
37                 login(request, user)
38                 return JsonResponse({'status': 200, 'message': 'Sesión iniciada',})
39             return JsonResponse({'status': 406, 'message': 'Contraseña incorrecta',})
40     json = {
41         'status': 200,
42         'campos': form_serializer(LoginForm())
43     }
44     return JsonResponse(json)
45
46 @api_login_required
47 def cerrar_sesion(request):
48     if request.method == 'POST':
49         logout(request)
50         return JsonResponse({'status': 200, 'message': 'Sesión cerrada',})
51     return JsonResponse({'status': 403, 'errors': 'No permitido',})
52
53
54 def create_usuario(request):
55     if request.method == 'POST':
56         form = UsuarioForm(request.POST)
57         if form.is_valid():
58             form.save()
59             return JsonResponse({'status': 201, 'message': 'Usuario creado',})
60
```

```
61         print(form.errors)
62         return JsonResponse({'status': 406, 'message': 'Error en los campos'})
63     json = {
64         'status': 200,
65         'campos': form_serializer(UsuarioForm())
66     }
67     return JsonResponse(json)
68
69
70 def create_dueño(request):
71     if request.method == 'POST':
72         form = DueñoForm(request.POST, request.FILES)
73         if form.is_valid():
74             form.save()
75             return JsonResponse({'status': 201, 'message': 'Dueño creado',})
76
77         print(form.errors)
78         return JsonResponse({'status': 406, 'message': 'Error en los campos'})
79     json = {
80         'status': 200,
81         'campos': form_serializer(DueñoForm())
82     }
83     return JsonResponse(json)
84
85
86 @api_login_required
87 def create_tienda(request):
88     if request.method == 'POST':
89         form = TiendaForm(request.POST)
90         if form.is_valid():
91             form.save()
92             return JsonResponse({'status': 201, 'message': 'Tienda creada',})
93
94         print(form.errors)
95         return JsonResponse({'status': 406, 'message': 'Error en los campos'})
96     json = {
97         'status': 200,
98         'campos': form_serializer(TiendaForm())
99     }
100     return JsonResponse(json)
101
102
103 @api_login_required
104 def get_tiendas(request):
105     if (request.user.tipo == 'DU'):
106         tiendas = Tienda.objects.filter(dueño=request.user)
107     else:
108         tiendas = Tienda.objects.all()
109     json = {
110         'status': 200,
111         'tipo': 'DU' if request.user.tipo == 'DU' else 'AD',
112         'tiendas': [model_serializer(tienda) for tienda in tiendas]
113     }
114     return JsonResponse(json)
115
116
```

```
117 @api_login_required
118 def create_producto(request):
119     if request.method == 'POST':
120         form = ProductoForm(request.POST)
121         if form.is_valid():
122             form.save()
123             return JsonResponse({'status': 200})
124
125     print(form.errors)
126     return JsonResponse({'status': 406, 'message': 'Error en los campos'})
127     json = {
128         'status': 200,
129         'campos': form_serializer(ProductoForm())
130     }
131     return JsonResponse(json)
132
133
134 @api_login_required
135 def get_productos(request):
136     productos = Producto.objects.filter(tienda=request.json()['tienda'])
137     json = {
138         'status': 200,
139         'productos': [model_serializer(producto) for producto in productos]
140     }
141     return JsonResponse(json)
142
143
144 @api_login_required
145 def create_venta(request):
146     if request.method == 'POST':
147         form = VentaForm(request.POST)
148         if form.is_valid():
149             form.save()
150             return JsonResponse({'status': 200})
151
152     return JsonResponse({'status': 406, 'message': 'Error en los campos'})
153     json = {
154         'status': 200,
155         'campos': form_serializer(VentaForm())
156     }
157     return JsonResponse(json)
158
159
160 @api_login_required
161 def get_ventas(request):
162     if request.user.tipo == 'US':
163         ventas = Venta.objects.filter(usuario=request.user)
164     else:
165         ventas = Venta.objects.filter(producto__tienda__dueño=request.user)
166     json = {
167         'status': 200,
168         'ventas': [model_serializer(venta) for venta in ventas]
169     }
170     return JsonResponse(json)
171
172 # ES DE PRUEBA, ELIMINAR EN PRODUCCION
```

```
173 def get_usuarios(request):
174     usuarios = Usuario.objects.all()
175     json = {
176         'status': 200,
177         'usuarios': [model_serializer(usuario) for usuario in usuarios]
178     }
179     return JsonResponse(json)
180
181 def get_errors(form_errors):
182     return [error for sublist in form_errors.values() for error in sublist]
```

- Se crean las vistas para los formularios de **Usuario**, **Dueño**, **Admin**, **Tienda**, **Producto** y **Venta**.
- También se crean las vistas para obtener los listados de los modelos de acuerdo al nivel de acceso.
- Estas vistas retornan un JSON con los datos del formulario tras ser serializados y reciben las respuestas del form.
- Se crean las vistas **iniciar_sesion** y **cerrar_sesion** para el inicio y cierre de sesión.

urls.py

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('forms_test/', views.forms_test, name='forms_test'),
6     path('iniciar_sesion/', views.iniciar_sesion, name='iniciar_sesion'),
7     path('cerrar_sesion/', views.cerrar_sesion, name='cerrar_sesion'),
8     path('create_usuario/', views.create_usuario, name='create_usuario'),
9     path('create_dueño/', views.create_dueño, name='create_dueño'),
10    path('create_tienda/', views.create_tienda, name='create_tienda'),
11    path('get_tiendas/', views.get_tiendas, name='get_tiendas'),
12    path('create_producto/', views.create_producto, name='create_producto'),
13    path('get_productos/', views.get_productos, name='get_productos'),
14    path('create_venta/', views.create_venta, name='create_venta'),
15    path('get_ventas/', views.get_ventas, name='get_ventas_dueño'),
16
17    path('get_usuarios/', views.get_usuarios, name='get_usuarios'), # ELIMINAR
18 ]
```

- Se crean las rutas para las vistas de los formularios y los listados de los modelos.
- Se crean las rutas para las vistas de inicio y cierre de sesión.

3.2 Frontend en Angular

3.2.1 Aplicación base

```
app.routes.ts
1 import { Routes } from '@angular/router';
2 import { DinamicFormComponent } from '../dinamic-form/dinamic-form.component';
3 import { HomeComponent } from '../home/home.component';
4 import { HeaderComponent } from '../home/header/header.component';
5 import { FooterComponent } from '../home/footer/footer.component';
6 import { LoginComponent } from '../login/login.component';
7 import { RegisterComponent } from '../register/register.component';
8
9 export const routes: Routes = [
10   {
11     path: '', component: HomeComponent
12   },
13   {
14     path: 'login', component: LoginComponent
15   },
16   {
17     path: 'register', component: RegisterComponent
18   }
19 ];
```

- Archivo de rutas de la aplicación.
- Se crean las rutas para los componentes **home**, **login** y **register**.

styles.css

```
1  /* You can add global styles to this file, and also import other style files */
2  html, body {
3    font-family: Arial, sans-serif;
4    margin: 0;
5    padding: 0;
6    display: flex;
7    flex-direction: column;
8  }
9
10 html, body {
11   height: 100%;
12 }
13
14 .container {
15   display: flex;
16   flex-direction: column;
17   min-height: 100vh;
18 }
19
20 header {
21 }
22
23 main {
24   flex: 1;
25   padding: 20px;
```

```
26 }
27
28 footer {
29     text-align: center;
30     padding: 10px 0;
31 }
```

- Estilos de la aplicación.
- Se encarga de dar estilos generales a la aplicación.

3.2.2 dynamic-form

```
                                loader-form.component.ts
1  import { Component, ComponentRef, ViewContainerRef } from '@angular/core';
2  import { DinamicFormComponent } from '../dinamic-form.component';
3  import { TYPE_FORMS } from '../../constants';
4
5  @Component({
6      selector: 'loader-form',
7      standalone: true,
8      imports: [],
9      template: `
10 `
11 })
12 export class LoaderFormComponent {
13     private componentRef!: ComponentRef<DinamicFormComponent>;
14
15     constructor(
16         private viewContainer: ViewContainerRef,
17     ){
18
19     }
20
21     createForm(typeForm: string): void {
22         if(this.componentRef)
23             this.componentRef.destroy();
24         this.componentRef = this.viewContainer.createComponent(DinamicFormComponent);
25         if(this.componentRef && this.componentRef.instance){
26             this.componentRef.instance.loadSchema(typeForm);
27         }else{
28             console.log("El componente no se ha podido cargar correctamente");
29         }
30     }
31
32 }
```

- Componente que se encarga de cargar el formulario dinámico.
- Se encarga de recibir los datos del formulario y enviarlos al componente **dynamic-form**.

dynamic-form.component.ts

```
1  import { Component } from '@angular/core';
2  import { FormBuilder, FormGroup, ReactiveFormsModule, Validators } from '@angular/forms';
3  import { FormField } from '../models/form-field';
```



```
4 import { ApiService } from '../services/api.service';
5
6 @Component({
7   selector: 'dinamic-form',
8   standalone: true,
9   imports: [ReactiveFormsModule],
10  templateUrl: './dinamic-form.component.html',
11  styleUrls: ['./dinamic-form.component.css']
12 })
13 export class DinamicFormComponent{
14
15   form: FormGroup;
16   fields: FormField[] = [];
17
18   constructor(
19     private formBuilder: FormBuilder,
20     private api: ApiService
21   ){
22     this.form = this.formBuilder.group({
23     });
24   }
25
26   loadSchema(model: string): void{
27     this.form = this.formBuilder.group({});
28     this.api.getFormSchema(model).subscribe(fieldsReceived => {
29       this.fields = fieldsReceived;
30       this.buildForm();
31     });
32   }
33
34   buildForm(){
35     for(const field of this.fields) {
36       const control = this.formBuilder.control('', this.getValidators(field));
37       this.form.addControl(field.name, control);
38     }
39   }
40
41   getValidators(field: FormField){
42     const validators = [];
43     if(field.validators){
44       for(const [key, value] of Object.entries(field.validators)){
45         switch(key){
46           case 'required':
47             validators.push(Validators.required); break;
48           case 'maxlength':
49             validators.push(Validators.maxLength(Number(value))); break;
50           case 'minlength':
51             validators.push(Validators.minLength(Number(value))); break;
52           case 'email':
53             validators.push(Validators.email); break;
54           default:
55             console.warn(`This validator is not supported: ${key}`);
56         }
57       }
58     }
59     return validators;
```

```

60 }
61
62 onSubmit(): void {
63     if (this.form.valid) {
64         console.log('Formulario enviado:', this.form.value);
65         // Enviar datos a la API
66     } else {
67         this.form.markAllAsTouched();
68         console.error('Formulario no válido');
69     }
70 }
71
72 getValidatorMessage(fieldName: string) {
73     const control = this.form.controls[fieldName];
74     if (control.errors) {
75         return Object.keys(control.errors)
76             .map(key => {
77                 switch (key) {
78                     case 'required':
79                         return 'Este campo es obligatorio';
80                     case 'minlength':
81                         return `Debe tener al menos ${control.errors?.['minlength'].requiredLength}
82                             ↪ caracteres`;
83                     case 'maxlength':
84                         return `Debe tener como máximo ${control.errors?.['maxlength'].requiredLength}
85                             ↪ caracteres`;
86                     case 'email':
87                         return 'Ingrese un email correcto';
88                     default:
89                         return '';
90                 }
91             })
92             .join(', ');
93     }
94     return '';
95 }

```

- Componente que se encarga de crear un formulario dinámico.
- Recibe un JSON con los campos del formulario y se encarga de deserializar los datos enviados desde Django.

dynamic-form.component.html

```

1  <form [formGroup]="form" (ngSubmit)="onSubmit()">
2    @for (field of fields; track $index) {
3      <label [for]="field.name">{{ field.label }}</label>
4      @switch (field.tipoCampo) {
5        @case ('input') {
6          <input [type]="field.attributes?.['type']"
7              [formControlName]="field.name"
8              [attr.placeholder]="field.attributes?.['placeholder'] || field.name"
9          >
10         }
11         @case ('select') {

```

```

12     <select [formControlName]="field.name"
13     >
14         <option value="" disabled selected>{{ field.attributes?.['placeholder'] ||
15             ↪ 'Seleccione una opción' }}</option>
16         @for (option of field.options; track $index) {
17             <option [value]="option.value">
18                 {{ option.label }}
19             </option>
20         }
21     </select>
22 }
23 @case ('textarea') {
24     <br>
25     <textarea [formControlName]="field.name"
26         [attr.placeholder]="field.attributes?.['placeholder'] || field.name"
27         [attr.rows]="field.attributes?.['rows'] || 10"
28         [attr.cols]="field.attributes?.['cols'] || 10"
29     ></textarea>
30 }
31 <br>
32
33 @if (form.get(field.name)?.invalid && form.get(field.name)?.touched) {
34     <div>
35         <small>
36             {{ getValidatorMessage(field.name) }}
37         </small>
38     </div>
39 }
40 }
41 <button>Guardar</button>
42 </form>

```

- Plantilla del componente **dynamic-form**.
- Se crean los campos del formulario de acuerdo al JSON recibido.

3.2.3 home

footer.component.ts

```

1  import { Component } from '@angular/core';
2  import { FaIconLibrary, FontAwesomeModule } from '@fortawesome/angular-fontawesome';
3  import { faFacebook, faInstagram, faWhatsapp } from '@fortawesome/free-brands-svg-icons';
4
5
6  @Component({
7      selector: 'home-footer',
8      standalone: true,
9      imports: [FontAwesomeModule],
10     template: `
11         <footer class="footer">
12             <div class="container-footer">
13                 <div class="row">
14                     <div>
15                         <h2 class="footer-heading logo">Kios Kios</h2>
16                         <p class="menu">

```

```
17         <a href="#">Home</a>
18         <a href="#">About</a>
19         <a href="#">Coments</a>
20         <a href="#">All</a>
21         <a href="#">Contact</a>
22     </p>
23     <ul class="ftco-footer-social ">
24         <li class="ftco-animate"><a href="#" data-toggle="tooltip"
25             ↳ data-placement="top" title="Whatsapp"><fa-icon [icon]="['fab',
26             ↳ 'whatsapp']"></fa-icon></a></li>
27         <li class="ftco-animate"><a href="#" data-toggle="tooltip"
28             ↳ data-placement="top" title="Facebook"><fa-icon [icon]="['fab',
29             ↳ 'facebook']"></fa-icon></a></li>
30         <li class="ftco-animate"><a href="#" data-toggle="tooltip"
31             ↳ data-placement="top" title="Instagram"><fa-icon [icon]="['fab',
32             ↳ 'instagram']"></fa-icon></a></li>
33     </ul>
34 </div>
35 </div>
36 </div>
37 </div>
38 <div class="row">
39     <p>
40         Kioskios 2024 | Sigue disfrutando de tus compras en KiosKios.com
41     </p>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
```

```
67
68 .footer .menu a:hover {
69     color: #fff;
70 }
71
72 .ftco-footer-social {
73     list-style: none;
74     padding: 0;
75     display: flex;
76     justify-content: center;
77     gap: 1em;
78 }
79
80 .ftco-footer-social li a {
81     display: block;
82     width: 40px;
83     height: 40px;
84     border: 1px solid #a3de83;
85     border-radius: 50%;
86     display: flex;
87     justify-content: center;
88     align-items: center;
89     transition: background 0.3s;
90 }
91
92 .ftco-footer-social li a:hover {
93     background: #a3de83;
94     color: #121212;
95 }
96
97 /* Responsivo */
98 @media (max-width: 768px) {
99     .footer .footer-heading {
100         font-size: 20px;
101     }
102     .footer .menu a {
103         font-size: 14px;
104     }
105 }
106
107 @media (max-width: 576px) {
108     .footer .footer-heading {
109         font-size: 18px;
110     }
111     .footer .menu a {
112         font-size: 12px;
113     }
114 }
115 })
116 export class FooterComponent {
117     constructor(library: FaIconLibrary) {
118         library.addIcons(faFacebook, faWhatsapp, faInstagram);
119     }
120 }
```

- Componente que se encarga de mostrar el footer de la página.

header.component.ts

```
1 import { Component } from '@angular/core';
2 import { RouterLink } from '@angular/router';
3 import { FaIconLibrary, FontAwesomeModule } from '@fortawesome/angular-fontawesome';
4 import { faCartShopping, faMagnifyingGlass } from '@fortawesome/free-solid-svg-icons';
5
6 @Component({
7   selector: 'home-header',
8   standalone: true,
9   imports: [RouterLink, FontAwesomeModule],
10  template: `
11    <header>
12      <div class="header-left">
13        <div class="logo">
14          
15        </div>
16        <div class="search">
17          <section class="search-bar">
18            <form action="." method="GET">
19              <label for="q"></label>
20              <input type="text" name="q" placeholder="Busca un artículo">
21              <button class="busqueda"><fa-icon [icon]='["fas", "magnifying-glass"]"></fa-icon></button>
22            </form>
23          </section>
24        </div>
25      </div>
26      <nav>
27        <ul>
28          <li><a [routerLink]="['/cart']"><fa-icon [icon]='["fas", "cart-shopping"]"></fa-icon> Carrito | </li>
29          <li><a [routerLink]="['/login']">Iniciar Sesión</a></li>
30          <li><a [routerLink]="['/register']" class="register">Regístrate</a></li>
31        </ul>
32      </nav>
33    </header>
34  `,
35  styles: `
36    /* Estilo del encabezado */
37    header {
38      display: flex;
39      justify-content: space-between;
40      align-items: center;
41      padding: 0.5em 5%;
42      background: #fff;
43      border-bottom: 1px solid #ddd;
44      position: sticky;
45      top: 0;
46      width: 100%;
47      z-index: 1000;
48      box-sizing: border-box;
49    }
50
51    header .logo img {
52      height: 60px;
53    }
54
55    .header-left{
```

```
56     display: flex;
57 }
58 nav ul {
59     list-style: none;
60     display: flex;
61     gap: 20px;
62 }
63
64 nav ul li a {
65     text-decoration: none;
66     color: #333;
67     padding: 0.5em 1em;
68     display: block;
69     white-space: nowrap;
70 }
71
72 nav ul li .register {
73     color: #ffffff;
74     background-color: #ff9900 ;
75     border-radius: 10%;
76     padding: 10px;
77 }
78
79 .busqueda{
80     color: #ffffff;
81     background-color: #ff9900 ;
82     border-radius: 10%;
83     padding: 5px;
84     border-color: #ff9900;
85 }
86
87 /* Barra de búsqueda */
88 .search{
89     justify-self: flex-start;
90 }
91 .search-bar {
92     text-align: center;
93     padding: 20px;
94     background: hsl(0, 0%, 100%);
95 }
96
97 .search-bar input {
98     padding: 10px;
99     width: 80%;
100     max-width: 500px;
101     border: 1px solid #e2e2e2;
102     border-radius: 4px;
103     width: 300px;
104     background-color: #ecec;
105     box-sizing: border-box;
106 }
107
108 /* Ajustes responsivos */
109 @media (max-width: 768px) {
110     header {
111         flex-direction: column;
```

```

112         align-items: flex-start;
113     }
114
115     .header-left {
116         width: 100%;
117         display: flex;
118         justify-content: space-between;
119         align-items: center;
120     }
121
122     nav ul {
123         flex-direction: column;
124         width: 100%;
125     }
126
127     nav ul li {
128         width: 100%;
129         text-align: center;
130     }
131
132     nav ul li a {
133         padding: 10px;
134         width: 100%;
135     }
136
137     .search-bar {
138         width: 100%;
139     }
140
141     .search-bar input {
142         width: calc(100% - 20px);
143     }
144 }
145 })
146 export class HeaderComponent {
147
148     constructor(library: FaIconLibrary){
149         library.addIcons(faCartShopping, faMagnifyingGlass);
150     }
151 }

```

- Componente que se encarga de mostrar el header de la página.

nav.component.ts

```

1  import { Component } from '@angular/core';
2
3  @Component({
4      selector: 'home-nav',
5      standalone: true,
6      imports: [],
7      template: `
8          <nav></nav>
9      `,
10     styles: ``
11 })

```



```
12 export class NavComponent {
13
14 }
```

- Componente que se encarga de mostrar la barra de navegación de la página.

banner.component.ts

```
1 import { Component, ElementRef, ViewChildren, QueryList } from '@angular/core';
2
3 @Component({
4   selector: 'home-banner',
5   standalone: true,
6   imports: [],
7   template: `
8     <section class="banner">
9       <div class="slider">
10         <div #slide class="slide active" style="background-image: url('banner.png');">
11           <div class="banner-text">
12             <h1>Revisa los nuevos productos disponibles</h1>
13             <p>Encuentra los mejores precios de tus productos y ahorra inteligentemente.</p>
14             <button>Registrate Ya</button>
15           </div>
16         </div>
17         <div #slide class="slide" style="background-image: url('banner2.png');justify-content:flex-end;">
18           <div class="banner-text" style="color: black">
19             <h1>Revisa los nuevos productos disponibles</h1>
20             <p>Encuentra los mejores precios de tus productos y ahorra inteligentemente.</p>
21             <button>Registrate Ya</button>
22           </div>
23         </div>
24
25         <div #slide class="slide" style="background-image: url('banner3.png');">
26           <div class="banner-text" style="color: black;">
27             <h1>Revisa los nuevos productos disponibles</h1>
28             <p>Encuentra los mejores precios de tus productos y ahorra inteligentemente.</p>
29             <button>Registrate Ya</button>
30           </div>
31         </div>
32       </div>
33       <div class="controls">
34         <button (click)="prevSlide()"></button>
35         <button (click)="nextSlide()"></button>
36       </div>
37     </section>
38 ` ,
39   styles: `/* Banner */
40   .banner {
41     position: relative;
42     overflow: hidden;
43     width: 100%;
44     height: 400px;
45   }
46
47   .slider {
48     display: flex;
```

```
49     position: relative;
50     flex-direction: column;
51     transition: transform 0.5s ease-in-out;
52 }
53
54 .slide {
55     position: absolute;
56     min-width: 100%;
57     height: 400px;
58     background-size: cover;
59     background-position: center;
60     opacity: 0;
61     top: 0;
62     left: 0;
63     transition: opacity .5s ease-in;
64 }
65
66
67
68 .slide.active { /* Mostrar la diapositiva activa */
69     opacity: 1;
70 }
71
72 @keyframes fade-in {
73     to {
74         opacity: 1;
75     }
76 }
77 @keyframes fade-out {
78     to {
79         opacity: 0;
80     }
81 }
82
83 @media (max-width: 768px) {
84     header {
85         flex-direction: column;
86         align-items: flex-start;
87     }
88
89     .header-left {
90         width: 100%;
91         display: flex;
92         justify-content: space-between;
93         align-items: center;
94     }
95
96     nav ul {
97         flex-direction: column;
98         width: 100%;
99     }
100
101     nav ul li {
102         width: 100%;
103         text-align: center;
104     }
```

```
105
106     nav ul li a {
107         padding: 10px;
108         width: 100%;
109     }
110
111     .search-bar {
112         width: 100%;
113     }
114
115     .search-bar input {
116         width: calc(100% - 20px);
117     }
118 }
119 .banner-text {
120     position: absolute;
121     top: 50%;
122     left: 10%;
123     transform: translateY(-50%);
124     color: #fff;
125     max-width: 400px;
126 }
127
128 .banner-text h1 {
129     font-size: 36px;
130     margin-bottom: 10px;
131 }
132
133 .banner-text p {
134     font-size: 18px;
135     margin-bottom: 20px;
136 }
137
138 .banner-text button {
139     padding: 10px 20px;
140     background: #ff9900;
141     border: none;
142     color: #fff;
143     cursor: pointer;
144     font-size: 16px;
145     border-radius: 4px;
146 }
147
148 /* Controles del banner */
149 .controls {
150     position: absolute;
151     top: 50%;
152     width: 100%;
153     display: flex;
154     justify-content: space-between;
155     transform: translateY(-50%);
156 }
157
158 .controls button {
159     background: rgba(0, 0, 0, 0.5);
160     color: #fff;
```

```
161     border: none;
162     padding: 10px;
163     cursor: pointer;
164     font-size: 20px;
165 }
166
167 .controls button#prevBtn {
168     margin-left: 10px;
169 }
170
171 .controls button#nextBtn {
172     margin-right: 10px;
173 }
174
175 /* Responsivo */
176 @media (max-width: 768px) {
177     .banner-text h1 {
178         font-size: 28px;
179     }
180
181     .banner-text p {
182         font-size: 16px;
183     }
184
185     .banner-text button {
186         padding: 8px 16px;
187         font-size: 14px;
188     }
189 }
190
191 })
192
193 export class BannerComponent {
194
195     @ViewChildren('slide') slides!: QueryList<ElementRef>;
196     private currentIndex: number = 0;
197
198     nextSlide() {
199         this.slides.toArray()[this.currentIndex].nativeElement.classList.remove('active');
200         this.currentIndex = (this.currentIndex + 1) % this.slides.length;
201         this.slides.toArray()[this.currentIndex].nativeElement.classList.add('active');
202     }
203
204     prevSlide() {
205         this.slides.toArray()[this.currentIndex].nativeElement.classList.remove('active');
206         this.currentIndex = (this.currentIndex - 1 + this.slides.length) % this.slides.length;
207         this.slides.toArray()[this.currentIndex].nativeElement.classList.add('active');
208     }
209 }
210
```

- Componente que se encarga de mostrar el banner de la página.

home.component.ts

```
1 import { Component } from '@angular/core';
2 import { HeaderComponent } from '../header/header.component';
```

```
3 import { NavComponent } from './nav/nav.component';
4 import { FooterComponent } from './footer/footer.component';
5 import { RouterOutlet } from '@angular/router';
6 import { BannerComponent } from './banner/banner.component';
7
8 @Component({
9   selector: 'app-home',
10  standalone: true,
11  imports: [HeaderComponent, NavComponent, FooterComponent, RouterOutlet, BannerComponent],
12  templateUrl: './home.component.html',
13  styleUrls: ['./home.component.css']
14 })
15 export class HomeComponent {
16
17 }
```

- Componente que se encarga de mostrar la página principal.
- Posee redirecciones a las demás páginas de la aplicación.

home.component.html

```
1 <home-header />
2 <home-nav />
3 <main>
4   <router-outlet />
5   <home-banner />
6 </main>
7 <home-footer />
8
```

- Plantilla del componente **home**.
- Se muestran los componentes **header**, **nav** y **footer**.

3.2.4 login

login.component.ts

```
1 import { AfterViewInit, Component, OnInit, ViewChild } from '@angular/core';
2 import { LoaderFormComponent } from '../dynamic-form/loader-form/loader-form.component';
3 import { TYPE_FORMS } from '../constants';
4
5
6
7 @Component({
8   selector: 'app-login',
9   standalone: true,
10  imports: [LoaderFormComponent],
11  templateUrl: './login.component.html',
12  styleUrls: ['./login.component.css']
13 })
14 export class LoginComponent implements AfterViewInit {
15
16   @ViewChild(LoaderFormComponent) loaderForm!: LoaderFormComponent;
17
18   ngAfterViewInit(): void {
```

```
19     if(this.loaderForm)
20         this.loaderForm.createForm(TYPE_FORMS.LOGIN);
21     else
22         console.log("No se pudo cargar el formulario de login");
23 }
24
25 }
```

- Componente que se encarga de mostrar el formulario de inicio de sesión.
- Se encarga de enviar los datos al backend y recibir la respuesta.

login.component.html

```
1 <main>
2   <loader-form />
3 </main>
```

- Plantilla del componente **login**.
- Se muestra el formulario de inicio de sesión.

3.2.5 register

register.component.ts

```
1 import { AfterViewInit, Component, ViewChild } from '@angular/core';
2 import { LoaderFormComponent } from '../dynamic-form/loader-form/loader-form.component';
3 import { TYPE_FORMS } from '../constants';
4 import { FooterComponent } from '../home/footer/footer.component';
5 import { HeaderComponent } from '../home/header/header.component';
6
7 @Component({
8   selector: 'app-register',
9   standalone: true,
10  imports: [LoaderFormComponent, FooterComponent, HeaderComponent],
11  templateUrl: './register.component.html',
12  styleUrls: ['./register.component.css']
13 })
14 export class RegisterComponent {
15
16   @ViewChild(LoaderFormComponent) loaderForm!: LoaderFormComponent;
17
18   loadOwner() : void {
19     this.loaderForm.createForm(TYPE_FORMS.CREATE_OWNER);
20   }
21
22   loadUser() : void {
23     this.loaderForm.createForm(TYPE_FORMS.CREATE_USER);
24   }
25
26 }
```

- Componente que se encarga de mostrar el formulario de registro.
- Se encarga de enviar los datos al backend y recibir la respuesta.

register.component.html

```
1 <home-header />
2 <br><br>
3 <main>
4   <div class="container">
5     <div class="buttons">
6       <button [(click)="loadOwner()" class="button" >Soy Dueño</button>
7       <button [(click)="loadUser()" class="button" >Soy Comprador</button>
8     </div>
9     <div class="form-container">
10      <loader-form />
11    </div>
12  </div>
13
14
15 </main>
16
17 <home-footer />
```

- Plantilla del componente **register**.
- Se muestra el formulario de registro.

register.component.css

```
1 .container{
2   display: flex;
3   flex-direction: column;
4   justify-content: flex-start;
5   align-items: center ;
6 }
7 .buttons{
8   display: flex;
9   flex-direction: row;
10  justify-content: center;
11  gap: 20px;
12  height: 100px;
13 }
14
15 .button{
16   color: #ffffff;
17   background-color: #ff9900 ;
18   border-radius: 10%;
19   padding: 10px;
20   width: 180px;
21   max-width: 180px;
22   max-height: 40px;
23   border-color: #ff9900;
24 }
25 .form-container {
26   background-color: rgb(233, 233, 233);
27   padding: 20px;
28   border-radius: 10px;
29   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
30     width: 100%;  
31     max-width: 400px;  
32 }
```

- Estilos del componente **register**.
- Se encarga de dar estilos al formulario de registro.

3.2.6 services

```
                                api.service.ts  
1  import { Injectable } from '@angular/core';  
2  import { HttpClient, HttpHeaders } from '@angular/common/http';  
3  import { Observable, catchError, map, of } from 'rxjs';  
4  import { FormField } from '../models/form-field';  
5  import { response } from 'express';  
6  import { error } from 'console';  
7  @Injectable({  
8    providedIn: 'root',  
9  })  
10 export class ApiService {  
11  
12   private urlBaseApi : string = 'http://localhost:8000/api';  
13  
14   constructor(private http: HttpClient) { }  
15  
16   getFormSchema(formToGet : string) : Observable<FormField[]> {  
17     const url = this.urlBaseApi + `/${formToGet}`;  
18     const httpOptions = {  
19       headers: new HttpHeaders({  
20         'Content-Type': 'application/json',  
21         //'Authorization': 'authkey',  
22         //'userid': '1'  
23       })  
24     };  
25     return this.http.get<{campos: FormField[]}>(url, httpOptions).pipe(  
26       map(response => {  
27         console.log("Response: ", response);  
28         if(!response.campos)  
29           throw new Error("No autorizado")  
30         return response.campos;  
31       }),  
32       catchError(error => {  
33         throw error;  
34       })  
35     );  
36   }  
37 }
```

- Servicio que se encarga de realizar las peticiones al backend.
- Se encarga de enviar los datos al backend y recibir la respuesta.

auth-interceptor.service.ts


```
1 import { HttpResponse, HttpEvent, HttpHandler, HttpInterceptor, HttpRequest } from
  ↳ '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { Router } from '@angular/router';
4 import { error } from 'console';
5 import { Observable, catchError } from 'rxjs';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class AuthInterceptorService implements HttpInterceptor {
11
12   constructor(private router: Router) {}
13
14   intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
15     return next.handle(req).pipe(
16       catchError((error: HttpResponse) => {
17         if(error.status === 401) {
18           const returnUrl = this.router.url;
19           this.router.navigate(['/login'], { queryParams: {returnUrl} })
20         }
21         throw error;
22       })
23     )
24   }
25 }
```

- Servicio que se encarga de interceptar las peticiones al backend.
- Se encarga de enviar el token de autenticación en las peticiones.

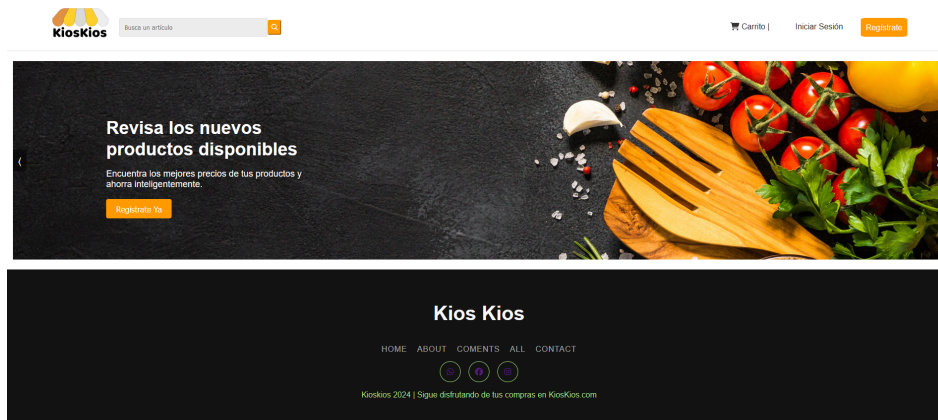
3.2.7 models

form-field.ts

```
1 export interface FormField {
2   tipoCampo: 'input' | 'select' | 'textarea';
3   name: string;
4   label: string;
5   validators?: { [key: string]: string };
6   attributes?: { [key: string]: string };
7   options?: { label: string; value: any }[];
8 }
```

- Modelo que se encarga de definir los campos de un formulario.
- Usado para la creación de formularios dinámicos.

3.3 Demostración



Página principal.

Nombre de usuario

Contraseña

Inicio de sesión.

Username

Telefono

Contraseña

Confirmar contraseña
















































































































Yape qr

Registro.











































































3.4 Lista de commits

Se manejan de forma más sólida las validaciones de los campos en el formulario	4854761	
guacreator committed last week		
Se añade un campo atributos opcional para los atributos de los elementos de entrada del formulario	5654f6a	
guacreator committed last week		
Se	18a1181	
guacreator committed last week		
Commits on Jun 29, 2024		
Merge branch 'main' of https://github.com/cmestaz/Kiosko	03a076d	
cmestaz committed last week		
Se abandona el soporte para windows	0b50916	
cmestaz committed last week		
Commits on Jun 28, 2024		
Merge pull request #4 from cmestaz/angularforms	894f094	
cmestaz committed last week		
Se reusa el componente para testarlo en la web, asimismo, se provee del HttpClient, con fetch ya que es recomendado para site server rendering	a76a887	
guacreator committed last week		
Se genera algunos campos de prueba para probar la funcionalidad de la construcción de formularios	3a65d9e	
guacreator committed last week		
Se construye en la estructura del componente el formulario con la construcción previa generada con los métodos en el componente	0a4a323	
guacreator committed last week		
Se crean los metodos del componente que permitieran construirlo con validaciones	0916361	
guacreator committed last week		
El componente FormBuilder fue reemplazado por el componente DynamicForm	47d2577	
guacreator committed last week		
Se crea el componente dynamicForm que construye los formularios que recibe de la api	0e48f1a	
guacreator committed last week		
Se modifica el modelo de los campos a recibir de la api para renderizar los formularios	2d90067	
guacreator committed last week		
Se modifican las vistas para retomar json y servir como api	03d3338	
cmestaz committed last week		
Merge pull request #3 from cmestaz/forms-api	8575c19	
cmestaz committed last week		
Se completa la funcionalidad de elegir una ubicación al momento de crear la tienda	7946033	
cmestaz committed last week		
Se solucionan los conflictos para añadir el campo de ubicación a la base de datos	8355238	
cmestaz committed last week		
Se modifica los modelos de usuarios para cumplir con las normas de la autenticación de django	46c5aea	
cmestaz committed last week		
Se modifican todos los modelos y formularios para ajustarse al sistema de autenticación de django	4d5c53a	
cmestaz committed last week		
Se crean formularios base para los modelos	c7a23a6	
cmestaz committed last week		
Merge pull request #2 from cmestaz/modelos	1726a33	
cmestaz committed last week		
Se añade un modelo dentro de los campos del formulario que permite colocar contenido en caso el elemento lo tenga	7d72796	
guacreator committed last week		
Merge pull request #1 from cmestaz/angularforms	3d0ca71	
cmestaz committed last week		
Se reea el servicio de la API con un método inicial para hacer peticiones de los formularios	3d4b081	
guacreator committed last week		
Se migran los modelos y añade una guía de instalación	5a53936	
cmestaz committed last week		
Se crea un modelo de campo del formulario para el envío de datos desde la api	6a06c98	
guacreator committed last week		
Creando el componente que construyó los formularios a partir de peticiones a la API	950a1a8	
guacreator committed last week		
Se crea la aplicación y los modelos	c11c9d0	
cmestaz committed last week		
Ignorando los archivos de configuración de vscode	a90a33f	
guacreator committed last week		
Se elimina la página por defecto de angular	8d7a48b	
guacreator committed last week		
Commits on Jun 26, 2024		
Instalación de los paquetes npm	8958b1c	
guacreator committed last week		
Añadiendo en el README la manera de instalar los requerimientos	35c38a0	
guacreator committed last week		
Commits on Jun 25, 2024		
Se inicializa el proyecto de angular	4803ae0	
cmestaz committed 2 weeks ago		
Se inicializa el proyecto de django	08aef0a	
cmestaz committed 2 weeks ago		
Se crean los requerimientos y el .gitignore	2463897	
cmestaz committed 2 weeks ago		
Inicial commit	2d48766	
cmestaz committed 2 weeks ago		

Lista de commits.

Se mueve el loader del formulario a un componente externo. Se crea un método en el mismo componente que permite cargar el formulario según un tipo.	626b7f51		
 guoCenator committed 4 days ago			
Se lanza una excepción cuando no se recibe una respuesta esperada de esquema de formulario para que el interceptor pueda manejar el error.	04ae325c		
 guoCenator committed 4 days ago			
Se crea un interceptor de las solicitudes para que redirija al login en caso el usuario no tenga los permisos para acceder a una parte de la página web	c803e10b		
 guoCenator committed 4 days ago			
Se cambia de nombre a las constantes para que sean más descriptivas	08b1c17a		
 guoCenator committed 4 days ago			
Se agrega una constante de formulario para el login	08a7c1d7		
 guoCenator committed 4 days ago			
Se cambia ligeramente el mensaje enviado en caso de error de autenticación	6c2d314		
 guoCenator committed 4 days ago			
Merge pull request #7 from cmetaszc/home	75f5304a		
 cmetaszc committed 4 days ago			
Merge branch 'main' into home	f4b758ac		
 cmetaszc committed 4 days ago			
Se realizan las peticiones a la api desde el servicio	e1a7f080		
 guoCenator committed 4 days ago			
Se utiliza la constante creada para abstraer los tipos de formularios que se obtendrán	5994c1e2		
 guoCenator committed 4 days ago			
Se modifica llamada a los archivos de acuerdo a la modificación del modelo del esquema del formulario	001910a		
 guoCenator committed 4 days ago			
Se simplifica el modelo del esquema de los formularios	1a1c405a		
 guoCenator committed 4 days ago			
Se crea un archivo de constantes para manejar de mejor manera las peticiones	5a6620a		
 guoCenator committed 4 days ago			
Se actualizan los requerimientos	c512257a		
 guoCenator committed 4 days ago			
Se cambia las respuestas de la api a las respuestas esperadas y un poco más adecuadas	0191137		
 guoCenator committed 4 days ago			
Se corrigen algunos problemas con el envío de la serialización	41097153		
 cmetaszc committed 4 days ago			
Merge pull request #6 from cmetaszc/forms-fix	70da38f		
 cmetaszc committed 4 days ago			
Se verifica el funcionamiento de los serializadores	007759df		
 cmetaszc committed 4 days ago			
Se añade la verificación del CORS	b01226d		
 cmetaszc committed 4 days ago			
Se crean serializadores para los modelos y formularios (no probado)	25c69f9		
 cmetaszc committed 4 days ago			
Se corrigen algunos errores de validación en los formularios	e5d4531		
 cmetaszc committed 4 days ago			
Se implementa el sistema de inicio de sesión y se añade un decorador para verificar si el usuario ha iniciado sesión	79b189d		
 cmetaszc committed 4 days ago			
Se corrige la subida de imágenes al almacenamiento	05a4b5e		
 cmetaszc committed 4 days ago			
Se deduce la integración de servicios de geolocalización en el backend	c998a527		
 cmetaszc committed 4 days ago			
 Commit on Jul 1, 2024			
Merge pull request #5 from cmetaszc/home	06776e6		
 cmetaszc committed 5 days ago			
Se cambian de campos para probar la funcionalidad del formulario del login	05a4b5e		
 guoCenator committed 5 days ago			
Se coloca el loader del en el componente del login	a0c8f06f		
 guoCenator committed 5 days ago			
Se elimina la carga de un formulario por defecto	2a8b5a1		
 guoCenator committed 5 days ago			
Se crea un cargador del formulario para que permita cargarlo dinámicamente. Luego lo referencia y le aplico el método que necesita para cargar de acuerdo al formulario requerido.	af120ca		
 guoCenator committed 5 days ago			
Se añade una ruta para el componente <code>register</code>	71a7a2a		
 guoCenator committed 5 days ago			
Se crea el componente <code>register</code> , que como su nombre lo indica, será para el registro de usuarios	4a67a76		
 guoCenator committed 5 days ago			
Se coloca un <code>router-outlet</code> en el componente principal y se delega los componentes de lo que llamamos la "página principal" al componente <code>main</code> . De esta manera, se podrán manejar las rutas adecuas...	9559c06		
 guoCenator committed 5 days ago			
Se añaden los componentes <code>nav</code> y <code>footer</code> al componente principal.	1a12a10		
 guoCenator committed 5 days ago			
Se une el componente header al home con el contenido que tenía anteriormente	711a5527		
 guoCenator committed 5 days ago			
 Commit on Jul 1, 2024			
Se crea un componente para la página principal llamado <code>main</code> y subcomponentes que son <code>header</code> , <code>nav</code> y <code>footer</code> . Adicionalmente, se creo un componente <code>login</code> para el inicio de sesión de los usuarios.	077c197a		
 guoCenator committed 5 days ago			
 Commit on Jun 30, 2024			
Se crea una estructura para la página principal de la aplicación, sin embargo, luego se componentizará la página en footer, header, nav, etc	4889717		
 guoCenator committed 5 days ago			

Lista de commits.

Se realizan unas mejoras al diseño. Posicion del header como sticky y slider principal con animación  guCraitor committed 45 hours ago	67a61a5 
Se envía la verificación del formulario del login en caso de que no sea de método post  guCraitor committed 1 hour ago	3c43a18 
Se mejora el diseño de la lupa de la sección de búsqueda  dmuelac1 committed 1 hour ago	68aea00 
Se completo el style del formulario de registro  dmuelac1 committed 2 hours ago	71a92a7 
Se implemento el diseño del contenedor del formulario  dmuelac1 committed 2 hours ago	8f55913 
Se estilizo el formulario de registro , falta terminar  dmuelac1 committed 2 hours ago	8d352ac 
Se convirtió el banner en un componente  dmuelac1 committed 14 hours ago	4d33275 
Se hizo el diseño del banner responsive  dmuelac1 committed 20 hours ago	3d6d680 
Se implemento mas componentes en el diseño del baner  dmuelac1 committed 20 hours ago	77a35d8 
Se implemento el diseño base de home del banner  dmuelac1 committed 20 hours ago	8f73985 
Se implemento el diseño base de home del banner  dmuelac1 committed 20 hours ago	4ac479f 
Se implemento la estructura html para el baner  dmuelac1 committed yesterday	6a772a8 
Se volvio responsive la barra de busqueda  dmuelac1 committed yesterday	8d5fcea 
Se realizaron modificaciones en el componente header para que no se desborde y sea adaptable  dmuelac1 committed yesterday → Commit on Jul 5, 2024	ac2088d 
Se importo icons de una libreria y se uso para los icons del footer  dmuelac1 committed yesterday	6a397d0 
Se mejora el style y se agrega componentes para que sea responsive  dmuelac1 committed yesterday	8851586 
Se implementa el style base del footer  dmuelac1 committed yesterday	15ff478 
Se agrega la estructura html con clases en el componente footer  dmuelac1 committed yesterday	e8f0a0c 
→ Commit on Jul 4, 2024	
Merge pull request #9 from cmestasz/home 	388a4c1 
 guCraitor committed 2 days ago	
Se regenera el archivo de requerimientos  cmestasz committed 2 days ago	4181c15 
Se hicieron algunas mejoras en el header  dmuelac1 committed 2 days ago	88f53ae 
Se estilizo la barra de busqueda del header  dmuelac1 committed 2 days ago	6d89f80 
Se agrega diseño a la sección de url de login , register en el header  dmuelac1 committed 2 days ago	89a8760 
Se agrega el diseño de la barra de header y el logo  dmuelac1 committed 2 days ago	61804ca 
Se realizo la estructura del header  dmuelac1 committed 2 days ago → Commit on Jul 3, 2024	5a6f73b 
Se añaden los requerimientos para Node en el README  guCraitor committed 3 days ago	2a7f70a 
Se crean las funciones que cargan los formularios de registro dependiendo de la persona que quiera registrarse  guCraitor committed 3 days ago	a31f488 
Se envía una cabecera en la petición con el formato de respuesta esperado  guCraitor committed 3 days ago	3c88043 
Se colocan botones que llaman funciones y que cargan el formulario dependiendo de cual se requiera  guCraitor committed 3 days ago	9284179 
El componente que crea los formularios ahora destruye los que ya ha creado. Asimismo, los construye solo cuando se le pide que lo cree.  guCraitor committed 3 days ago	12a0582 
Se revisó como está enviada la validación de email y se colocó en el apartado de validaciones  guCraitor committed 3 days ago	7f46371 
Se coloca al servidor de Angular como origen permitido para hacer peticiones a la api  guCraitor committed 3 days ago → Commit on Jul 2, 2024	3a4ea29 
Merge branch 'main' of https://github.com/cmestasz/kioticias  cmestasz committed 4 days ago	378a326 
Se añaden estados a las vistas para encajar con el frontend de angular  cmestasz committed 4 days ago	395294c 
Merge pull request #8 from cmestasz/home 	77891a8 
 cmestasz committed 4 days ago	
→ Commit on Jul 2, 2024	
En el login se creó el formulario del login. Aun no se ha implementado la capacidad de redirigir o de hacer submit.  guCraitor committed 4 days ago	4a8228d 

Lista de commits.

4 URL del repositorio en GitHub

- <https://github.com/cmestasz/KiosKios.git>

5 Enlace al video

- <https://youtu.be/uV5e1j8fDv0>

6 Estructura de laboratorio 10

El contenido que se entrega en este laboratorio es el siguiente:

```
1 KiosKios/
2 |--- Informe.pdf
3 |--- Informe.tex
4 |--- README.md
5 |--- kioskios_api
6     |--- api
7         |--- __init__.py
8         |--- admin.py
9         |--- apps.py
10        |--- decorators.py
11        |--- forms.py
12        |--- migrations
13        |--- models.py
14        |--- serializers.py
15        |--- tests.py
16        |--- urls.py
17        |--- views.py
18    |--- kioskios_api
19        |--- __init__.py
20        |--- settings.py
21        |--- urls.py
22        |--- wsgi.py
23    |--- manage.py
24 |--- kioskios_web
25     |--- src
26         |--- app
27             |--- app.routes.ts
28             |--- app.component.html
29             |--- app.component.css
30             |--- dinamic-form
31                 |--- loader-form
32                     |--- loader-form.component.ts
33             |--- dynamic-form
34                 |--- dynamic-form.component.ts
35                 |--- dynamic-form.component.html
36                 |--- dynamic-form.component.spec.ts
37                 |--- dynamic-form.component.css
38         |--- home
39             |--- footer
40                 |--- footer.component.ts
41             |--- header
42                 |--- header.component.ts
43             |--- nav
```

```
44         |--- nav.component.ts
45     |--- banner
46         |--- banner.component.ts
47     |--- home.component.ts
48     |--- home.component.html
49     |--- home.component.css
50 |--- login
51     |--- login.component.ts
52     |--- login.component.html
53     |--- login.component.css
54 |--- register
55     |--- register.component.ts
56     |--- register.component.html
57     |--- register.component.css
58 |--- services
59     |--- api.service.ts
60     |--- auth-interceptor.service.ts
61 |--- models
62     |--- form-field.ts
63 |--- (otros archivos de configuración)
64 |--- index.html
65 |--- styles.css
66 |--- main.ts
67 |--- main.server.ts
68 |--- (otros archivos de configuración)
69 |--- img
70     |--- logo_abet.png
71     |--- logo_episunsa.png
72     |--- logo_unsa.jpg
73     |--- E1.png
74     |--- E2.png
75     |--- E3.png
76     |--- commits1.png
77     |--- commits2.png
78     |--- commits3.png
```

7 Referencias

- <https://angular.dev/tutorials/learn-angular>
- <https://www.djangoproject.com/>