Manual Técnico del Proyecto: “Alumnalia” Gestión de Datos Personales y Profesionales

[1. Introducción](#_heading=h.gvpts4l7lz1l) 2

[2. Configuración Inicial](#_heading=h.couuazmuvfbq) 2

[3. Modelos de Datos](#_heading=h.giksraci9a8u) 3

[3.1 Dat\_Per (Datos Personales)](#_heading=h.pkv4h5134g3n) 3

[3.2 Inf\_Prof (Información Profesional)](#_heading=h.o92tcl4ndpeo) 3

[4. Funcionalidades del Proyecto](#_heading=h.qv59s3o32ytf) 4

[5. Pruebas](#_heading=h.97gem5scv11d) 4

[6. Despliegue](#_heading=h.n2viffd92qlm) 4

[7. Mantenimiento y Actualización](#_heading=h.a1s838gz0aig) 5

[8. Documentación Adicional](#_heading=h.c6ctao6uxxtl) 5

[9. Formularios](#_heading=h.uafalbvw0hhl)**5**

[9.1. Formulario DatPerForm](#_heading=h.n8x3lzqh9y4t) 5

[9.1.1. Campos](#_heading=h.5an4vwyjeaah) 5

[9.1.2. Etiquetas](#_heading=h.a45hr1v3rot1) 6

[9.1.3. Widgets](#_heading=h.klbflmhey051) 7

[9.2. Formulario FormadoresForm](#_heading=h.o5ioz98xkh6k) 7

[9.2.1. Campos](#_heading=h.t3hepxu36xbi) 7

[9.2.2. Etiquetas](#_heading=h.2d9ly7j5crdg) 8

[9.2.3. Widgets](#_heading=h.tvr8p7qrwoue) 9

[9.3. Formulario InfoEstuForm](#_heading=h.gwelrkl6a04e) 9

[9.3.1. Campos](#_heading=h.z2z4sfci5xn5) 9

[9.3.2. Etiquetas](#_heading=h.dxsdyowoe9cv) 9

[9.3.3. Widgets](#_heading=h.gf6smye8kl9o) 10

[9.4. Configuración de URLs y Vistas](#_heading=h.mk4nwhe9zc5r)10

[9.4.1. Configuración de URLs](#_heading=h.x94wpcwul6gj) 10

[9.4.2.Descripción de las Rutas](#_heading=h.nw9ww3jd2hw1) 11

[9.5. Configuración de Vistas](#_heading=h.3i4uj411mvri) 13

[9.5.1. Importaciones Necesarias](#_heading=h.u15speeduboi) 13

[9.5.2. Vistas Basadas en Clases](#_heading=h.a1rzlzt9nksu) 13

[9.6. Estilos y Temas del Proyecto (Base CSS)](#_heading=h.vbrt0psfeq4b) 17

[9.6.1. Estructura del CSS](#_heading=h.vzjymw6ckmrh) 18

# 1. Introducción

* **Descripción del Proyecto:** Este proyecto implementa una aplicación web para la gestión de datos personales y profesionales utilizando Django como framework.
* **Objetivo:** Facilitar la gestión y el almacenamiento seguro de información sensible, como datos personales, información profesional, competencias y certificaciones.

# 2. Configuración Inicial

* **Requisitos del Sistema:** Descripción de las dependencias necesarias y cómo instalarlas (Python, Django, etc.).
* **Instalación:** Instrucciones paso a paso para clonar el repositorio, configurar el entorno virtual e instalar las dependencias.

Git clone <repositorio>

cd <repositorio>

python3 -m venv venv

source venv/bin/actívate

pip install -r requirements.txt

# 3. Modelos de Datos

## **3.1 Dat\_Per (Datos Personales)**

* **Campos:**
  + pk\_per: Identificador único.
  + nom\_per: Nombre.
  + dni\_per: DNI/NIE.
  + fn\_per: Fecha de nacimiento.
  + cn\_per: Nacionalidad.
  + tel\_per: Teléfono.
  + email\_per: Correo electrónico.
  + dir\_per: Dirección.
  + sex\_per: Género.
  + uso\_datos\_per, term\_per, noti\_per: Aceptación de términos y consentimientos.
* **Validaciones:**
  + validar\_dni: Verifica que el DNI tenga 9 caracteres alfanuméricos.
  + validar\_longitud\_nueve: Verifica que el número de teléfono tenga exactamente 9 dígitos.
* **Encriptación:** Se encriptan los datos sensibles antes de guardarlos.

self.nom\_per = encrypt\_data(self.nom\_per)

def get\_nom\_per(self):

return decrypt\_data(self.nom\_per)

## **3.2 Inf\_Prof (Información Profesional)**

* **Campos:**
  + pk\_inf\_pro: Identificador único.
  + tit\_inf\_pro: Título académico más alto obtenido.
  + esp\_inf\_pro: Área de especialización principal.
  + exp\_inf\_pro: Años de experiencia como formador/a.
  + for\_imp\_inf\_pro: Tipo de formación impartida.
  + cv\_adj\_inf\_pro: Currículum en formato PDF.
  + cert\_inf\_pro: Certificaciones docentes.
  + herr\_inf\_pro: Herramientas tecnológicas utilizadas.
  + comp\_dig\_inf\_pro: Competencias digitales específicas.
  + mod\_inf\_pro: Modalidades de enseñanza preferidas.
  + tipo\_alu\_inf\_pro: Tipo de alumnado preferido.
  + franja\_inf\_pro: Franjas horarias disponibles.
  + fk\_per\_inf\_pro: Relación con Dat\_Per.

# 4. Funcionalidades del Proyecto

* **CRUD de Datos Personales y Profesionales:** Descripción de las operaciones de Crear, Leer, Actualizar y Eliminar para cada modelo.
* **Encriptación y Desencriptación:** Detalles sobre cómo se manejan los datos sensibles.
* **Validaciones:** Explicación de las validaciones realizadas para garantizar la integridad de los datos.

# 5. Pruebas

* **Pruebas Unitarias:** Instrucciones para ejecutar pruebas unitarias que verifiquen la funcionalidad de los modelos y métodos.

python manage.py test

# 6. Despliegue

* **Configuración de Producción:** Instrucciones para configurar el entorno de producción, incluyendo ajustes de seguridad y optimización.
* **Despliegue en el Servidor:** Pasos para desplegar la aplicación en un servidor web.

# 7. Mantenimiento y Actualización

* **Guía de Mantenimiento:** Instrucciones para mantener y actualizar la aplicación.
* **Manejo de Errores Comunes:** Soluciones para problemas comunes que pueden surgir durante el uso de la aplicación.

# 8. Documentación Adicional

* **Referencias:** Cualquier documentación adicional que sea relevante para el proyecto.
* **Manual de Usuario:** Guía para los usuarios finales sobre cómo utilizar la aplicación.

# 9. Formularios

## **9.1. Formulario DatPerForm**

El formulario DatPerForm se utiliza para manejar los datos personales de los usuarios. Este formulario está basado en el modelo Dat\_Per.

### **9.1.1. Campos**

* **nom\_per**: Nombre de la persona.
* **dni\_per**: DNI/NIE de la persona.
* **fn\_per**: Fecha de nacimiento.
* **cn\_per**: Nacionalidad.
* **tel\_per**: Número de teléfono.
* **email\_per**: Dirección de correo electrónico.
* **fk\_nom\_via**: Tipo de vía (relación con la tabla TipoVia).
* **dir\_per**: Dirección de la persona.
* **fk\_nom\_pro**: Provincia (relación con la tabla Provincias).
* **fk\_nom\_mun**: Municipio (relación con la tabla Municipios).
* **sex\_per**: Género de la persona.
* **uso\_datos\_per**: Consentimiento para uso de datos con fines de gestión académica y administrativa.
* **term\_per**: Aceptación de términos y condiciones.
* **noti\_per**: Deseo de recibir notificaciones de oportunidades formativas.

### **9.1.2. Etiquetas**

Las etiquetas son las descripciones legibles por humanos que se muestran junto a los campos del formulario.

labels = {

'nom\_per': 'Nombre de la Persona',

'dni\_per': 'DNI',

'fn\_per': 'Fecha de Nacimiento',

'cn\_per': 'Nacionalidad',

'tel\_per': 'Teléfono',

'email\_per': 'Correo Electrónico',

'fk\_nom\_via': 'Tipo de Vía',

'dir\_per': 'Dirección',

'fk\_nom\_pro': 'Provincia',

'fk\_nom\_mun': 'Municipio',

'sex\_per': 'Género',

'uso\_datos\_per': 'Consentimiento para Uso de Datos',

'term\_per': 'Aceptación de Términos y Condiciones',

'noti\_per': 'Desea Recibir Notificaciones'

}

### **9.1.3. Widgets**

#### Los widgets personalizan la apariencia y el comportamiento de los campos del formulario.

widgets = {

'fn\_per': forms.DateInput(attrs={'type': 'date'}),

'uso\_datos\_per': forms.CheckboxInput(),

'term\_per': forms.CheckboxInput(),

'noti\_per': forms.CheckboxInput(),

'fk\_nom\_pro': forms.Select(),

'fk\_nom\_mun': forms.Select(),

'fk\_nom\_via': forms.Select()

}

## **9.2. Formulario FormadoresForm**

El formulario FormadoresForm se utiliza para manejar la información profesional de los formadores. Este formulario está basado en el modelo Inf\_Prof.

### **9.2.1. Campos**

* **tit\_inf\_pro**: Título académico más alto obtenido.
* **tit\_esp\_inf\_pro**: Especificar título.
* **esp\_inf\_pro**: Área de especialización principal.
* **exp\_inf\_pro**: Años de experiencia como formador/a.
* **for\_imp\_inf\_pro**: Tipo de formación impartida.
* **for\_imp\_esp\_inf\_pro**: Especificar el tipo de formación impartida.
* **mod\_inf\_pro**: Modalidades de enseñanza preferidas.
* **tipo\_alu\_inf\_pro**: Tipo de alumnado preferido.
* **tipo\_alu\_esp\_inf\_pro**: Especificar tipo de alumnado.
* **franja\_inf\_pro**: Franjas horarias disponibles.
* **cert\_inf\_pro**: Certificación docente.
* **cert\_esp\_inf\_pro**: Especificar certificación.
* **herr\_inf\_pro**: Herramientas tecnológicas utilizadas.
* **herr\_esp\_inf\_pro**: Especificar herramientas.
* **comp\_dig\_inf\_pro**: Competencias digitales específicas (ej. DigCompEdu).
* **comp\_dig\_esp\_inf\_pro**: Especificar competencias digitales.
* **fk\_per\_inf\_pro**: Relación con la tabla Dat\_Per.
* **cv\_adj\_inf\_pro**: Adjuntar currículum en formato PDF.

### **9.2.2. Etiquetas**

labels = {

'tit\_inf\_pro': 'Título académico más alto obtenido',

'tit\_esp\_inf\_pro': 'Especificar título',

'esp\_inf\_pro': 'Área de especialización principal',

'exp\_inf\_pro': 'Años de experiencia como formador/a',

'for\_imp\_inf\_pro': 'Tipo de formación impartida',

'for\_imp\_esp\_inf\_pro': 'Especificar el tipo de formación impartida',

'mod\_inf\_pro': 'Modalidades de enseñanza preferidas',

'tipo\_alu\_inf\_pro': 'Tipo de alumnado preferido',

'tipo\_alu\_esp\_inf\_pro': 'Especificar tipo de alumnado',

'franja\_inf\_pro': 'Franjas horarias disponibles',

'cert\_inf\_pro': 'Certificación docente',

'cert\_esp\_inf\_pro': 'Especificar certificación',

'herr\_inf\_pro': 'Herramientas tecnológicas utilizadas',

'herr\_esp\_inf\_pro': 'Especificar herramientas',

'comp\_dig\_inf\_pro': 'Competencias digitales específicas (ej. DigCompEdu)',

'comp\_dig\_esp\_inf\_pro': 'Especificar competencias digitales',

'fk\_per\_inf\_pro': 'Datos de personas',

'cv\_adj\_inf\_pro': 'Adjunta tu currículum en formato PDF'}

### **9.2.3. Widgets**

widgets = {

'cv\_adj\_inf\_pro': forms.FileInput(attrs={'accept': '.pdf'}),}

## **9.3. Formulario InfoEstuForm**

El formulario InfoEstuForm se utiliza para manejar la información de los estudiantes. Este formulario está basado en el modelo Info\_Estu.

### **9.3.1. Campos**

* **int\_form\_inf\_est**: Intereses formativos.
* **que\_que\_est**: Qué querrían estudiar.
* **que\_has\_est**: Qué han estudiado.
* **de\_que\_han\_trab**: De qué han trabajado.
* **de\_que\_que\_trab**: De qué querrían trabajar.
* **rang\_sal\_des**: Rango salarial deseado.
* **cv\_adj\_inf\_est**: Adjuntar currículum en formato PDF.
* **fk\_per\_inf\_est**: Relación con la tabla Dat\_Per.

### **9.3.2. Etiquetas**

labels = {

'int\_form\_inf\_est': 'Intereses Formativos',

'que\_que\_est': 'Qué Querrían Estudiar',

'que\_has\_est': 'Qué Han Estudiado',

'de\_que\_han\_trab': 'De Qué Han Trabajado',

'de\_que\_que\_trab': 'De Qué Querrían Trabajar',

'rang\_sal\_des': 'Rango Salarial Deseado',

'cv\_adj\_inf\_est': 'Adjunta tu Currículum en Formato PDF',

'fk\_per\_inf\_est': 'Datos de Personas'

}

## **9.3.3. Widgets**

widgets = {

'rang\_sal\_des': forms.NumberInput(),

'cv\_adj\_inf\_est': forms.FileInput(attrs={'accept': '.pdf'}),

}

## **9.4. Configuración de URLs y Vistas**

#### **Descripción**

Este apartado describe la configuración de URLs y vistas en el proyecto Django de Alumnalia. La configuración de URLs permite definir las rutas de acceso a las diferentes vistas del proyecto.

### **9.4.1. Configuración de URLs**

#### **Estructura del Archivo urls.py**

El archivo urls.py principal se encarga de incluir las rutas a las distintas vistas del proyecto. A continuación, se presenta la estructura y el contenido del archivo:

python

from django.contrib import admin

from django.urls import path, include, get\_resolver

from . import views

from django.conf.urls.static import static

from django.conf import settings

app\_name = "app\_alumnalia"

urlpatterns = [

path("", views.InicioView.as\_view(), name='inicio'),

path("inicio/", views.InicioView.as\_view(), name='inicio'),

# path('\_\_debug\_\_/', include('debug\_toolbar.urls')),

path('perfil\_usuario/', views.PerfilusuarioView.as\_view(), name='perfil\_usuario'),

path("datos\_personales\_estudiantes/", views.datos\_personales\_estudiantes\_view.as\_view(), name='datos\_personales\_estudiantes'),

path("datos\_personales\_formadores/", views.datos\_personales\_formadores\_view.as\_view(), name='datos\_personales\_formadores'),

path('datos\_formador/', views.datos\_formador\_view.as\_view(), name='datos\_especificos\_formador'),

path('datos\_estudiante/', views.datos\_estudiante\_view.as\_view(), name='datos\_especificos\_estudiante'),

path('exito/', views.exito\_view.as\_view(), name='exito'),

path("oferta\_personalizada/", views.oferta\_personalizada.as\_view(), name='oferta\_personalizada'),

path('admin/', admin.site.urls),

# path('eliminar/<int:id>/', views.eliminar\_fila, name='eliminar\_fila'), # Elimina un Dato de Dat\_Per

# path('filas/', views.lista\_filas, name='lista\_filas'),

]

# print(f" control --> {get\_resolver().reverse\_dict.keys()} <")

#### 

### **9.4.2.Descripción de las Rutas**

* **Inicio**:

path("", views.InicioView.as\_view(), name='inicio')

path("inicio/", views.InicioView.as\_view(), name='inicio')

Define las rutas para la vista de inicio del proyecto.

* **Perfil de Usuario:**

path('perfil\_usuario/', views.PerfilusuarioView.as\_view(), name='perfil\_usuario')

Ruta para la vista del perfil de usuario.

* **Datos Personales de Estudiantes**:

path("datos\_personales\_estudiantes/", views.datos\_personales\_estudiantes\_view.as\_view(), name='datos\_personales\_estudiantes')

Ruta para la vista de datos personales de estudiantes.

* **Datos Personales de Formadores**:

path("datos\_personales\_formadores/", views.datos\_personales\_formadores\_view.as\_view(), name='datos\_personales\_formadores')

Ruta para la vista de datos personales de formadores.

* **Datos Específicos del Formador**:

path('datos\_formador/', views.datos\_formador\_view.as\_view(), name='datos\_especificos\_formador')

Ruta para la vista de datos específicos del formador.

* **Datos Específicos del Estudiante**:

path('datos\_estudiante/', views.datos\_estudiante\_view.as\_view(), name='datos\_especificos\_estudiante')

Ruta para la vista de datos específicos del estudiante.

* **Vista de Éxito**:

path('exito/', views.exito\_view.as\_view(), name='exito')

Ruta para la vista de confirmación de éxito.

* **Oferta Personalizada**:

path("oferta\_personalizada/", views.oferta\_personalizada.as\_view(), name='oferta\_personalizada')

Ruta para la vista de oferta personalizada.

* **Administración**:

path('admin/', admin.site.urls)

Ruta para la vista de administración.

### **9.5. Configuración de Vistas**

Las vistas en Django son fundamentales para definir la lógica que se ejecuta cuando un usuario accede a una URL específica. En este proyecto, utilizamos una combinación de vistas basadas en clases y vistas basadas en funciones para manejar las diversas necesidades de la aplicación.

### **9.5.1. Importaciones Necesarias**

import base64

from django.shortcuts import render, redirect

from django.http import HttpResponseRedirect

from .forms import \*

from .models import \*

from django.views.generic import FormView, TemplateView, View

from django.shortcuts import render, get\_object\_or\_404, redirect

Se importan los módulos y funciones necesarios para manejar las vistas y formularios.

### **9.5.2. Vistas Basadas en Clases**

#### **Vista InicioView**

Define la vista para la página de inicio de la aplicación.

class InicioView(TemplateView):

template\_name = "app\_alumnalia/inicio.html"

context\_object\_name = ""

##### **Vista PerfilusuarioView**

Define la vista para el perfil de usuario.

class PerfilusuarioView(TemplateView):

template\_name = "app\_alumnalia/perfilusuario.html"

context\_object\_name = ""

##### **Vista oferta\_personalizada**

Define la vista para mostrar ofertas personalizadas.

class oferta\_personalizada(TemplateView):

template\_name = "app\_alumnalia/ofertassugeridas.html"

context\_object\_name = ""

##### **Vista datos\_personales\_estudiantes\_view**

Maneja la visualización y el procesamiento del formulario de datos personales de los estudiantes.

class datos\_personales\_estudiantes\_view(View):

template\_name = "app\_alumnalia/datospersonalesestudiantesform.html"

def get(self, request):

form = DatPerForm()

return render(request, self.template\_name, {'form': form})

def post(self, request):

form = DatPerForm(request.POST)

if form.is\_valid():

form.save()

return redirect('oferta\_personalizada')

return render(request, self.template\_name, {'form': form})

##### **Vista datos\_personales\_formadores\_view**

Maneja la visualización y el procesamiento del formulario de datos personales de los formadores.

class datos\_personales\_formadores\_view(View):

template\_name = "app\_alumnalia/datospersonalesformadoresform.html"

def get(self, request):

form = DatPerForm()

return render(request, self.template\_name, {'form': form})

def post(self, request):

form = DatPerForm(request.POST)

if form.is\_valid():

form.save()

return redirect('datos\_formador')

return render(request, self.template\_name, {'form': form})

##### **Vista datos\_formador\_view**

Maneja la visualización y el procesamiento del formulario de datos específicos del formador.

class datos\_formador\_view(FormView):

template\_name = "app\_alumnalia/formadoresform.html"

def get(self, request, \*args, \*\*kwargs):

form = FormadoresForm()

return self.render\_to\_response({'form': form})

def post(self, request, \*args, \*\*kwargs):

form = FormadoresForm(request.POST, request.FILES) # Asegúrate de incluir request.FILES

if form.is\_valid():

form.save()

return redirect('exito') # Redirigir después de guardar

return self.render\_to\_response({'form': form})

##### **Vista datos\_estudiante\_view**

Maneja la visualización y el procesamiento del formulario de datos específicos del estudiante.

class datos\_estudiante\_view(FormView):

template\_name = 'app\_alumnalia/estudiantesform.html'

def get(self, request, \*args, \*\*kwargs):

form = InfoEstuForm()

return self.render\_to\_response({'form': form})

def post(self, request, \*args, \*\*kwargs):

form = InfoEstuForm(request.POST, request.FILES)

if form.is\_valid():

form.save()

return redirect('exito')

return self.render\_to\_response({'form': form})

##### **Vista exito\_view**

Define la vista para la página de éxito después de completar un formulario.

class exito\_view(View):

template\_name = 'app\_alumnalia/exito.html'

def get(self, request):

return render(request, self.template\_name)

## **9.6. Estilos y Temas del Proyecto (Base CSS)**

##### **Descripción de los Templates**

En el proyecto Alumnalia, se prestó especial atención al diseño visual, incluyendo el tipo de letra y los colores, para asegurar una experiencia de usuario consistente y atractiva. A continuación, se describe cómo se implementaron estos aspectos en el archivo de estilos CSS.

#### **Tipografía**

* **Fuente Principal**: Segoe UI
* **Fuente Secundaria**: system-ui, -apple-system

Estas fuentes se aplican a través de la declaración global en el CSS:

body, html {

height: 100%;

font-family: 16px system-ui, -apple-system, "Segoe UI";

color: #0736f8;

background-color: white !important;

#### }

#### **Colores**

La paleta de colores definida para el proyecto incluye:

* Color Principal: #0736f8 (Azul Klean)
* Color de Texto Principal: #374c5d (Gris)
* Color de Fondo: #ffffff (Blanco)

#### 

### **9.6.1. Estructura del CSS**

#### **Reglas Generales**

Se aplica un enfoque de diseño adaptable con box-sizing para todos los elementos:

\* {

box-sizing: border-box;

}

#### **Contenedores de Imágenes con Texto de Fondo**

.image-container {

display: flex;

justify-content: space-around;

align-items: center;

flex-wrap: wrap;

margin: 20px 0;

z-index: 10;

}

.image-box {

position: relative;

width: 45%;

height: 500px;

margin: 10px;

}

.bg-image {

width: 100%;

height: 100%;

filter: blur(6px);

background-position: center;

background-repeat: no-repeat;

background-size: cover;

position: absolute;

top: 0;

left: 0;

z-index: 1;

}

.bg-text {

background-color: rgba(0, 0, 0, 0.4);

color: #e5f4f4;

font-weight: bold;

font-family: system-ui, -apple-system, "Segoe UI", sans-serif;

border: 3px solid #e5f4f4;

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

z-index: 2;

width: 80%;

padding: 20px;

text-align: center;

}

#### **Formularios**

Estilos para formularios y alineación de etiquetas y entradas:

.form {

background-color: #07976c;

font-family: 16px system-ui, -apple-system, "Segoe UI";

color: #374c5d;

}

.form-group {

display: flex;

align-items: center;

justify-content: space-between;

}

.form-group label {

flex: 1;

}

.form-group input, .form-group select {

flex: 2;

}

#### **Navegación y Carrusel**

header .col-md-3 {

display: flex;

align-items: center;

}

header .nav {

align-items: center;

}

img {

margin-left: 15px;

}

.nav-link {

padding: 10px 15px;

}

.carousel-inner {

width: 100%;

height: 100%;

}

.carousel-inner img {

width: 100%;

height: 100%;

}

.carousel-item {

height: 500px;

}

.carousel-caption {

background-color: rgba(0, 0, 0, 0.5);

}

.carousel {

position: relative;

margin: 50px;

}

button {

background-color: #31b0b5;

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 18px;

padding: 6px 24px;

border-radius: 12px;

}