

Proyecto Final de Grado

**Ciclo Formativo de Grado Superior en Desarrollo de
Aplicaciones Web**

Título:

DonaCrypto: Plataforma de Donaciones Descentralizadas para Creadores de Contenido

Autor:

Joshua Núñez Arcila

Centro:

Grupo Studio Formación

Fecha:

Junio 2025

Índice

| | |
|---|--|
| 1. Abstract | |
| 2. Introducción | |
| 2.1 Contexto del proyecto | |
| 2.2 Justificación de la idea | |
| 2.3 Objetivos | |
| 3. Análisis y planificación | |
| 3.1 Requisitos funcionales | |
| 3.2 Requisitos no funcionales | |
| 3.3 Estudio de viabilidad | |
| 3.4 Herramientas y tecnologías utilizadas | |
| 4. Diseño | |
| 4.1 Arquitectura de la aplicación | |
| 4.2 Estructura del sistema | |
| 4.3 Diseño de la base de datos | |
| 5. Desarrollo e implementación | |
| 5.1 Funcionalidades desarrolladas | |
| 6. Conclusiones y mejoras futuras | |
| 6.1 Logros alcanzados | |
| 6.2 Dificultades encontradas | |
| 6.3 Posibles mejoras o ampliaciones | |
| 6.4 Conclusión Final | |

1. Abstract

DonaCrypto es una plataforma web que permite a creadores de contenido recibir donaciones en criptomonedas de manera directa, segura y sin intermediarios. El objetivo principal es empoderar a los creadores, eliminando comisiones y barreras geográficas, y garantizando la privacidad tanto de donantes como de creadores.

La plataforma ofrece un sistema de gestión de perfiles para creadores y seguidores, un feed personalizado, gestión de múltiples wallets de criptomonedas, y un sistema de publicaciones tipo red social.

El desarrollo de DonaCrypto responde a la necesidad de alternativas descentralizadas frente a plataformas tradicionales de donaciones, motivado por la búsqueda de mayor transparencia, control y libertad financiera para los usuarios.

2. Introducción

2.1 Contexto del proyecto

En la economía digital actual, los creadores de contenido dependen de plataformas centralizadas para monetizar su trabajo, lo que implica comisiones elevadas, restricciones geográficas y pérdida de privacidad. Las criptomonedas y la tecnología blockchain abren la puerta a modelos más justos y descentralizados.

2.2 Justificación de la idea

La descentralización permite que los creadores reciban el 100% de las donaciones, sin intermediarios ni comisiones. Además, la privacidad y la libertad de operar globalmente son ventajas clave frente a sistemas tradicionales.

2.3 Objetivos

Objetivo general:

Desarrollar una plataforma web descentralizada que facilite donaciones directas en criptomonedas a creadores de contenido.

Objetivos específicos:

- Permitir a los creadores gestionar su perfil y wallets de criptomonedas.
- Ofrecer a los seguidores un feed personalizado y herramientas para descubrir y apoyar creadores.
- Garantizar la privacidad y seguridad de las transacciones.
- Implementar un sistema de publicaciones y seguimiento entre usuarios.

3. Análisis y planificación

3.1 Requisitos funcionales

- Registro y autenticación de usuarios (creador/seguidor)
- Gestión de perfiles y avatares
- Gestión de wallets (BTC, ETH, USDT, etc.)
- Sistema de posts y feed personalizado
- Seguimiento y descubrimiento de creadores
- Donaciones directas a wallets
- Búsqueda de creadores por username
- Responsive design

3.2 Requisitos no funcionales

- Seguridad en el manejo de datos y autenticación (JWT)
- Escalabilidad y modularidad del sistema
- Facilidad de uso y accesibilidad
- Compatibilidad multiplataforma

3.3 Estudio de viabilidad

- Técnica: Uso de tecnologías modernas (Vue.js, Flask, MongoDB) y APIs públicas de blockchain.
- Económica: Sin costes de licencias, infraestructura local y open source.
- Operativa: Interfaz intuitiva, soporte multi-idioma y documentación clara.

3.4 Herramientas y tecnologías utilizadas

- **Frontend:** Vue.js 3, Composition API, Pinia, SASS, Vite
- **Backend:** Flask, PyMongo, JWT, Python Logging

- **Base de datos:** MongoDB

4. Diseño

4.1 Arquitectura de la aplicación

Arquitectura cliente-servidor RESTful.

Separación clara entre frontend (SPA) y backend (API REST).

Uso de JWT para autenticación y autorización.

4.2 Estructura del sistema

```
backend/
├── app.py                # Punto de entrada de la aplicación
├── app/
│   ├── __init__.py      # Factory pattern para app Flask
│   ├── extensions.py    # Configuración de extensiones
│   ├── models/          # Modelos de datos
│   │   ├── user.py      # Modelo de usuario
│   │   ├── post.py      # Modelo de posts
│   │   ├── wallet.py    # Modelo de wallets
│   │   └── following.py  # Modelo de seguimiento
│   ├── routes/           # Endpoints de la API
│   │   ├── auth_routes.py # Autenticación (/auth/*)
│   │   └── user_routes.py  # Usuarios y creadores (/user/*)
│   └── utils/            # Utilidades
│       └── validators.py  # Validaciones personalizadas
├── requirements.txt      # Dependencias Python
└── tests/               # Tests automatizados
```

```

frontend/
├── src/
│   ├── App.vue           # Componente raíz
│   ├── main.js           # Punto de entrada
│   ├── assets/           # Assets estáticos
│   │   └── styles/
│   │       └── style.scss # Sistema de diseño Bento UI
│   ├── components/       # Componentes reutilizables
│   │   ├── auth/
│   │   │   └── AuthComponent.vue
│   │   └── common/
│   │       └── PostCard.vue
│   ├── stores/           # Estado global (Pinia)
│   │   ├── auth.js       # Store de autenticación
│   │   └── dashboard.js  # Store del dashboard
│   ├── views/            # Vistas principales
│   │   ├── auth/         # Login/Registro
│   │   ├── creator/      # Dashboard creador
│   │   ├── follower/     # Feed y perfil follower
│   │   ├── HomeView.vue  # Página principal
│   │   └── AboutView.vue # Acerca de
│   ├── router/           # Configuración de rutas
│   │   └── index.js
│   └── services/         # Servicios HTTP
│       └── api.js        # Cliente API con Axios
├── package.json          # Dependencias Node.js
└── vite.config.js        # Configuración de Vite

```

4.3 Diseño de la base de datos

Modelo entidad-relación simple:

Colección: users

```
{
  "_id": ObjectId,
  "email": String (unique, required),
  "username": String (unique, required),
  "password": String (bcrypt hash, required),
  "role": String (enum: ["creator", "follower"]),
  "bio": String (optional),
  "avatar_url": String (optional),
  "created_at": Date,
  "updated_at": Date
}
```

Colección: wallets

```
{
  "_id": ObjectId,
  "user_email": String (foreign key),
  "currency_type": String (enum: ["BTC", "ETH", "USDT"]),
  "wallet_address": String (required),
  "is_default": Boolean,
  "created_at": Date
}
```

Colección: posts

```
{
  "_id": ObjectId,
  "creator_email": String (foreign key),
  "title": String (required),
  "content": String (required),
  "media_urls": Array[String] (optional),
  "likes_count": Number (default: 0),
  "created_at": Date,
  "updated_at": Date
}
```

Colección: following

```
{
  "_id": ObjectId,
  "follower_email": String (foreign key),
  "creator_email": String (foreign key),
  "created_at": Date
}
```


5. Desarrollo e implementación

5.1 Descripción de funcionalidades desarrolladas

- Registro y login con roles (creador/seguiror)
- Gestión de perfil: edición de datos, avatar, contraseña
- Gestión de wallets: agregar, editar, eliminar, QR, múltiples criptomonedas
- Sistema de posts: crear, eliminar, feed personalizado, likes, compartir
- Seguimiento: seguir/dejar de seguir, lista de seguidos, búsqueda de creadores
- Explorador de creadores: búsqueda por username, exploración global
- Responsive design: interfaz adaptable a móviles y escritorio
- Notificaciones: feedback visual con Toast
- Seguridad: autenticación JWT, validación de formularios

6. Conclusiones y mejoras futuras

6.1 Logros alcanzados

- Plataforma funcional y estable para donaciones descentralizadas
- Experiencia de usuario intuitiva y moderna
- Seguridad y privacidad garantizadas
- Sistema modular y escalable

6.2 Dificultades encontradas

- Serialización de fechas BSON/ISO entre backend y frontend
- Manejo de estados y sincronización en tiempo real
- Validación y feedback de formularios

6.3 Posibles mejoras o ampliaciones

- Soporte para más criptomonedas y tokens
- Conversión automática de precios y alertas
- Sistema de notificaciones en tiempo real (WebSockets)
- Integración con wallets externas (Metamask, WalletConnect)
- Gamificación (badges, rankings de donantes)
- Internacionalización y soporte multi-idioma
- App móvil

6.4 Conclusión Final:

Se ha desarrollado exitosamente una plataforma web que cumple con todos los objetivos planteados. El sistema permite a creadores gestionar su presencia digital y monetización de manera descentralizada, mientras que los seguidores pueden descubrir, seguir e interactuar con contenido de calidad.

DonaCrypto representa una evolución natural en la economía de creadores, aprovechando las ventajas de la tecnología blockchain para crear un ecosistema más justo, transparente y libre. La plataforma no solo cumple con los objetivos técnicos planteados, sino que establece las bases para un futuro donde los creadores tienen control total sobre su monetización y audiencia.