

# DOCUMENTO BASE DE ARQUITECTURA

Tuya Crowd Funding

**Revisión:**

1.0

## HISTORIAL DE CAMBIOS

Fecha	Descripción	Realizado por	Revisión
20/02/2022	Creación del documento	Sebastián Urrego Juan Pablo Gómez Daniel Giraldo Diego Guzmán Andrés Almánzar	1.0

# CONTENIDOS

## 1. ALCANCE

Se hará una plataforma para Tuya, que tiene como objetivo realizar hacer usuarios inversionistas que puedan depositar el dinero en la plataforma ya sea como una donación o una inversión como accionista de Tuya.

Se acordó con el cliente de Tuya a la hora de realizar ciertas acciones como el login, o los tokens que se generan al invertir o que no es necesario implementar una pasarela de pagos, además de que se deben simular las bases de datos sin ninguna conexión a una API o un componente externo de Tuya.

Debemos de tener en cuenta ciertos limites tributarios para verificar si se esta haciendo un lavado de activos, pero de la misma mano de las inversiones y de los otros atributos del sistema muchas de estas cosas serán simulados y no se basarán en datos reales.

Como objetivo principal es crear una aplicación o plataforma en la que puedan hacer estas acciones que se nos piden, todo con datos simulados y crear ambientes ficticios para poder hacer la prueba de la aplicación, teniendo también en cuenta que las tecnologías a utilizar son .NET, Angular y mongoDB como base de datos.

## 2. DESCRIPCIÓN DE COMPONENTES

A la hora de mirar los componentes tenemos que dividirlos en diferentes secciones:

### **Presentación:**

Esta la basaremos en Angular, manejando con ella todas las vistas de la aplicación creando diferentes componentes para cada una de las vistas, se crearán componentes para:

- El componente de los contribuyentes
- El componente de las donaciones
- El componente de los proyectos
- El componente de los chats
- El componente de los mensajes
- El componente de los empresarios

Cada uno con sus respectivas validaciones y diseños que nos ayudaran a tener una mejor experiencia con el usuario. Esta se conecta con los usuarios en sus navegadores y con peticiones HTTP se les devuelve un Response en HTTP. Se comunican con la lógica del negocio a través de REST/JSON para el envío de datos.

### **Lógica de negocio:**

Esta se hará en .NET, con este conectaremos la capa de presentación con la persistencia de datos, la idea es realizar dos divisiones la cual serían los modelos que estarán muy basados en el diagrama

de clases que se tiene y los controladores que serán los que manejan el tráfico entre la persistencia de datos y la interfaz de usuario. En esta capa estará toda la lógica de negocio de la aplicación.

#### **Persistencia de datos:**

El guardado de la información se hará en una base de datos NoSQL en MongoDB, basaremos los modelos de datos muy similares a los modelos que construyamos en .NET, aquí se guardara todos los datos relacionados con la información de los usuarios y las donaciones que se tengan.

### **3. DIMENSIÓN DEL SISTEMA**

A la hora de dimensionar un sistema debemos de tener en cuenta tres casos de tiempo, el optimista, el pesimista y el óptimo. De esta forma creemos que podemos dar una respuesta más asertiva a las validaciones de estimaciones ya que tendríamos una buena forma de simular las dimensiones del sistema de todo tipo.

En cuestiones de tiempo de respuesta esperemos en el peor de los casos 30 segundos de demora en las transacciones ya que podemos implementar modelos incorrectos en la capa de datos o hacer consultas extremadamente pesados, pero si todo esto lo implementamos bien, esperamos tiempos de respuesta de menos de 5 segundos por transacción, pero siendo óptimos y con la cantidad de datos que vamos a manejar diría que el promedio de las transacciones esta en los 10 segundos.

La cantidad de usuarios esperada es amplia esperamos realizar un tipo de chat en línea entre pares por lo que esperemos una cantidad de usuarios mayor a dos al mismo tiempo, pero verdaderamente esperemos que se puedan realizar acciones al mismo tiempo sin afectar las transacciones de los demás por lo que esperamos una cantidad de usuarios aproximados de 50.000 usuarios intentando escalar a los 2.6 millones de usuarios que se encuentran en el mercado colombiano.

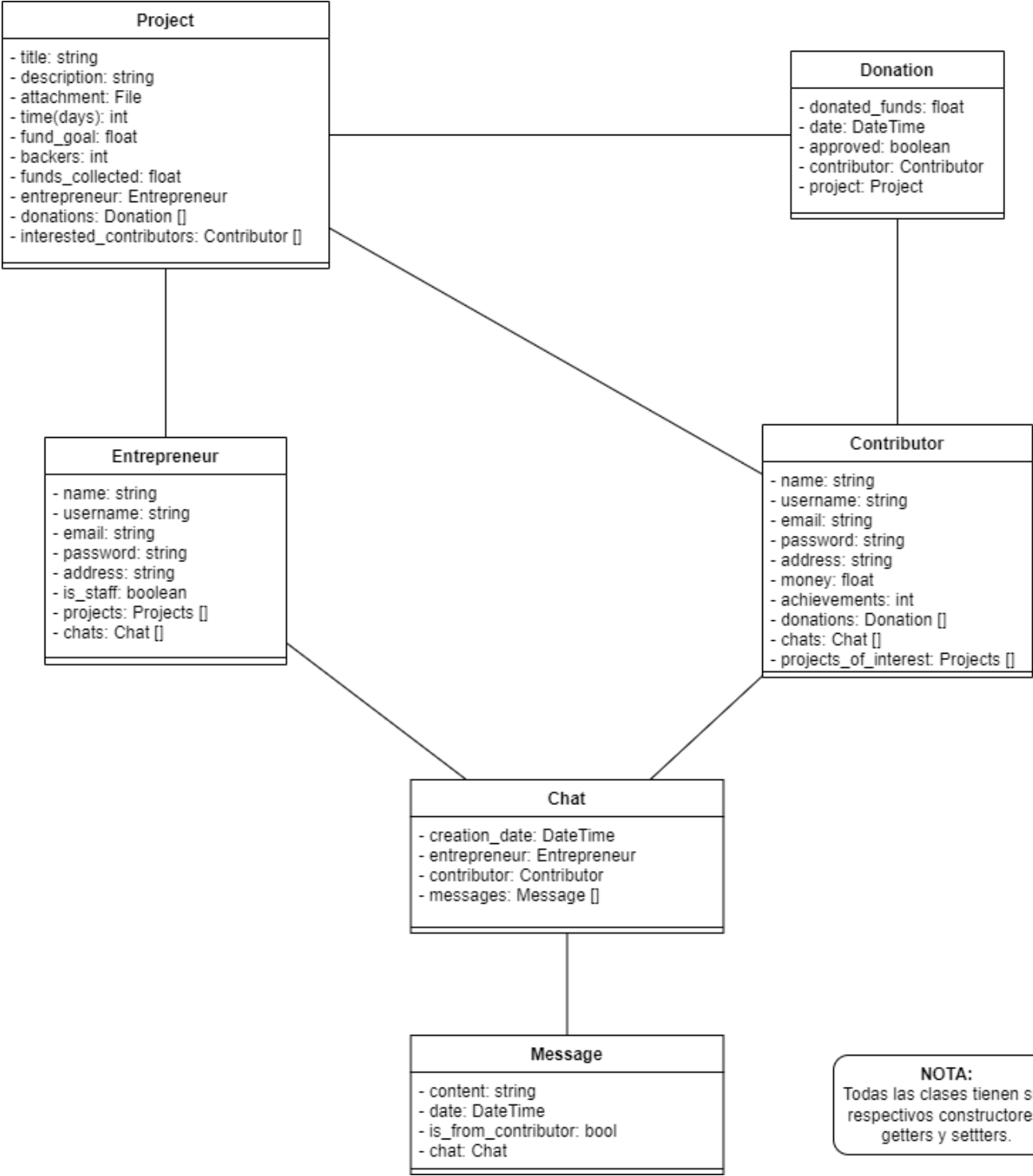
Con esta cantidad de usuarios esperamos una cantidad inmensa de peticiones en el día, incluso un usuario podría llegar a más de 200 peticiones al día ya sea con chats o donaciones por lo que esperamos una cantidad casi igual que el numero de usuarios que utilice la plataforma, porque también depende de cuanta cantidad de donaciones o ingresos a la plataforma hagan al día, pero teniendo en cuenta la cantidad de clientes de tuya, diría que tal vez unas 70.000 peticiones con 50.000 usuarios al día.

### **4. OTROS REQUISITOS NO FUNCIONALES**

[RNF01] Que sea posible observar el producto en cualquier dispositivo.

[RNF02] Manejar la misma paleta de colores en todos los elementos y componentes de la aplicación.

# DIAGRAMA DE CLASES



# DIAGRAMA DE COMPONENTES DEL SISTEMA

