

UNIVERSIDAD POLITÉCNICA DE CATALUÑA

FACULTAD DE INFORMÁTICA DE BARCELONA

INGENIERÍA DE SOFTWARE

Repositorio de árboles genealógicos en BD NoSQL

Autor:

Daniel ALBARRAL NUÑEZ

Director:

Enric MAYOL



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

1. Resumen/Abstract

Índice

1. Resumen/Abstract	1
2. Introducción	2
3. Alcance	2
4. Planificación	2
5. Presupuesto y impacto social	2
6. Análisis de requisitos	2
6.1. Visión global del sistema	2
6.2. Actores	2
6.3. Requisitos funcionales	3
6.3.1. Usuario final	3
6.3.2. Aplicación	3
6.4. Requisitos no funcionales	4
7. Especificación	5
8. Diseño	5
9. Implementación	5
10.Pruebas	5
11.Conclusiones	5
12.Referencias	5

2. Introducción

3. Alcance

4. Planificación

5. Presupuesto y impacto social

6. Análisis de requisitos

6.1. Visión global del sistema

Como ya se ha comentado en los capítulos anteriores, el proyecto consiste en el desarrollo de una API RESTful desde la que poder gestionar un repositorio de árboles genealógicos.

Para ello el *backend* implementado deberá permitir que diferentes aplicaciones se den de alta para poder acceder a los datos almacenados por los usuarios y así mismo permitir a estos usuarios realizar las operaciones necesarias para gestionar sus árboles genealógicos.

6.2. Actores

Los actores implicados en el uso del sistema son dos:

Usuario final :

Este actor será el que se dará de alta en nuestra aplicación y ha de ser capaz de almacenar y gestionar sus árboles genealógicos.

Aplicación :

El *backend* ha de permitir a diferentes aplicaciones (*frontends*) darse de alta.

Estas aplicaciones recibirán las credenciales necesarias para poder conseguir los permisos de acceso de nuestros usuarios finales y manipular sus árboles.

6.3. Requisitos funcionales

Los requisitos funcionales describen todas las funcionalidades que implementara el *backend*. A continuación se listan todos los requisitos funcionales desglosados por actor:

6.3.1. Usuario final

Gestión de cuenta :

Registro : El usuario deberá poderse registrar en la aplicación introduciendo los datos necesarios.

Modificar sus datos : El usuario deberá poder modificar sus datos.

Login : El usuario deberá de poder acceder al una interfaz desde la que gestionar sus datos de usuario.

Activación de cuenta : El usuario una vez registrado tendrá que poder activar la cuenta vía email.

6.3.2. Aplicación

Gestión de cuenta :

Registro : El usuario deberá poderse registrar en la aplicación introduciendo los datos necesarios.

Login : El usuario deberá de poder acceder al una interfaz desde la que gestionar sus datos de usuario.

Modificar sus datos : El usuario deberá poder modificar sus datos.

Gestión de aplicaciones :

Dar de alta aplicación : El usuario ha de poder dar de alta aplicaciones que posteriormente accederán al contenido del repositorio mediante la API.

Conseguir token : Las aplicaciones una vez dadas de alta podrán pedir *tokens* asignados a los usuarios finales con su nivel de autorización que permita su autenticación.

Gestión de los árboles :

Creación, modificación y consulta de árboles genealógicos : El usuario deberá poderse crear, modificar y consultar los árboles genealógicos.

Creación, modificación y consulta de personas : El usuario deberá de poder crear, modificar y consultar las personas almacenadas en los árboles genealógicos.

Creación, modificación y consulta de eventos : El usuario deberá de poder crear, modificar y consultar los eventos de las personas almacenadas en los árboles genealógicos.

Subir un archivo GEDCOM : El usuario deberá de poder cargar archivos GEDCOM en su cuenta.

Encontrar personas similares : El usuario deberá de poder seleccionar una persona de uno de sus árboles genealógicos y buscar personas similares en otros árboles de otros usuarios.

6.4. Requisitos no funcionales

A continuación se muestra una lista de los requisitos no funcionales del sistema:

1. La API ha de permitir un uso completo independientemente de la tecnología que la use.
2. Se ha de proporcionar una interfaz para gestionar los usuarios de la aplicación (No se ha de hacer a través de endpoints de la API).

3. El tiempo de respuesta de todos los endpoints ha de ser similar sea cuan sea la funcionalidad que implementen.
4. La plataforma ha de constar de sistemas que aseguren la seguridad de los datos de usuario.
5. Todos los *endpoints* de la API tendrán que respetar los estandartes de una aplicación rest ful

7. Especificación

8. Diseño

9. Implementación

10. Pruebas

11. Conclusiones

12. Referencias