

MANUAL TÉCNICO

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE PARA LA GESTIÓN
DE HISTORIAS CLÍNICAS

JEFFREY STEVEN GARCÍA GALLEGO
JOSUÉ PEÑA ATENCIO
MAURICIO CORTÉS DÍAZ

PONTIFICIA UNIVERSIDAD JAVERIANA SECCIONAL CALI
INGENIERÍA DE SISTEMAS
CALI, COLOMBIA
2020

1. Presentación

El siguiente manual guiará al personal encargado del soporte y mantenimiento del sistema, el cual les dará a conocer los requerimientos y la estructura para la construcción del sistema, en el desarrollo de programa de escritorio conectado mediante una base de datos proporcionada por Firebase.

2. Objetivo

Informar y especificar al usuario la estructura y conformación del sistema con el fin de que pueda hacer soporte y modificaciones o actualizaciones al sistema en general.

3. Requisitos del sistema

- Requerimientos de hardware
Equipo, teclado, mouse, monitor.
Memoria RAM 2GB
- Requerimientos de software (servidor)
Sistema operativo (Ubuntu 18.02 LTS)
Python 3.0.
Compilador de C.
Conexión a internet.

4. Herramientas utilizadas para el desarrollo

Python (Django)

Es un lenguaje de programación interpretado cuya filosofía se centra en la legibilidad del código. Es un lenguaje multiparadigma, soporta orientación a objetos, programación imperativa y en menor medida, programación funcional. Es el lenguaje de programación el cual usa *Django*, el cual es un framework de desarrollo web de código abierto. Se caracteriza por su fácil manejo y versatilidad para proyectos.

Firebase Realtime Database y Firebase Auth

El servicio de Realtime Database consiste en una base de datos en tiempo real, que vendría a denominarse ‘back-end’, la cual maneja una estructura en forma de árbol JSON. Este servicio proporciona a los desarrolladores de aplicaciones una API para permitir la sincronización y almacenamiento de datos en los servidores de Firebase.

Por otro lado, *Firebase Auth* es un servicio para autenticación de usuarios para realizar inicios de sesión, entre otras funcionalidades relacionadas.

Pyrebase

Es un proyecto en Github el cual trata de hacer un wrap en lenguaje Python para la API que ofrece Firebase, haciendo el uso de esta aún más sencillo y cómo para desarrolladores con poca experiencia en el manejo de APIs.

5. Instalación de aplicaciones

Requisitos generales pre-instalación del sistema

Para ejecutar el programa es necesario haber instalado ciertos paquetes en *Python*. Los cuales son fácilmente instalados a través de este sencillo comando.

```
D:\
λ pip install django pyrebase|
```

Las configuraciones de los archivos estáticos, las aplicaciones y las rutas que debe seguir Django para desplegar la aplicación se encuentran en el archivo *settings.py*.

```
hc_manager > settings.py
11
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = '#u^@1a5_p+ro^y^=dp6y4bhp0+73h2sw17xfhm-%zmj@hllra'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = ['*']
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'login.apps.LoginConfig',
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
```

Para el manejo de los archivos estáticos si se está utilizando alguna herramienta para el despliegue es necesario realizar modificaciones a las rutas de los archivos estáticos. Estas, también están presentes en el archivo *settings.py*.

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC_URL = "/static/"
STATIC_DIRS = (
    os.path.join(BASE_DIR, 'static')
)
```

6. Acceso a la base de datos

La aplicación debe configurarse para que pueda conectarse con la base de datos de *Firebase Realtime*. A través de las siguientes líneas de códigos se establece la conexión entre la aplicación y la base de datos.

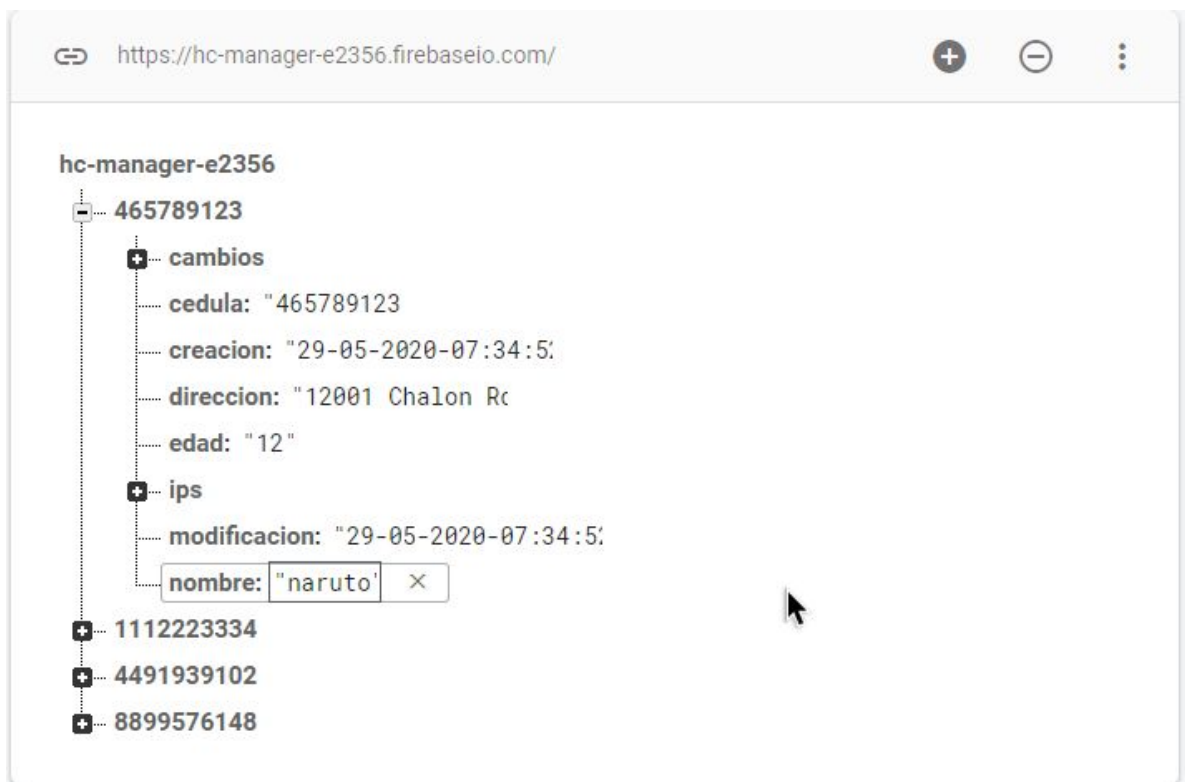
```
from django.shortcuts import render
from datetime import datetime
import pyrebase
import json

# Configuración del wrap pyrebase
config = {
    'apiKey': "AIzaSyBeSa2PSyHEtt9jPRYZXjRP4myOvG8XoGc",
    'authDomain': "hc-manager-e2356.firebaseio.com",
    'databaseURL': "https://hc-manager-e2356.firebaseio.com",
    'projectId': "hc-manager-e2356",
    'storageBucket': "hc-manager-e2356.appspot.com",
    'messagingSenderId': "869491009077",
    'appId': "1:869491009077:web:35468c3d8127791a55337c",
    'measurementId': "G-C33NC6SJC8"
}

# Inicialización de servicios de la base de datos
firebase = pyrebase.initialize_app(config)
```

La información en el diccionario config puede variar dependiendo de si se está usando el mismo proyecto o no.

En la consola de *Firebase Realtime* podemos hacer seguimiento de los datos y generar interacción con estos. Así como también operaciones básicas sobre los datos (CRUD).



7. Copias de seguridad y restauración en la base de datos

Firebase no ofrece este servicio de manera gratuita, por lo tanto, si se desea obtener esta funcionalidad es necesario realizar un desembolso a través de la consola de Firebase y seguir los pasos allí estipulados.

8. Referencias

Django: <https://www.djangoproject.com/>

Pyrebase: <https://github.com/thisbejim/Pyrebase>

Firebase: <https://firebase.google.com/>