

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACION DE TECNOLOGOS

DESARROLLO DE APLICACIÓN WEB Y MÓVIL DEL “CENTRO DE DESARROLLO INFANTIL DESCUBRIR” PARA NOTIFICACION E INCIDENCIAS DEL SERVICIO DEL TRANSPORTE ESCOLAR Y SEGUIMIENTO ESCOLAR

Manual de Instalación

Josselyn Denisse Vela Nieto

josselyn.vela@epn.edu.ec

Bryan Geovanny Farinango Buse

bryan.farinango@epn.edu.ec

DIRECTOR: ING.JUAN PABLOZALDUMBIDE PROAÑO, MSC.

juan.zaldumbide@epn.edu.ec

CODIRECTOR: PhD. Diana Cecilia Yacchirema Vargas

diana.yacchirema@epn.edu.ec

Quito, septiembre 2021

Índice de Contenido

| | |
|--|----------|
| 1. Despliegue del sistema web en Digital Ocean | 3 |
| 1.1 Creación de un nuevo proyecto en Firebase | 3 |
| 1.2 Conexión de Firebase con Angular | 3 |
| 1.3 Creación de la base de datos en MongoDB | 4 |
| 1.4 Despliegue del Front-end | 6 |
| 1.5 Despliegue del Back-end | 7 |
| 2. Despliegue de la aplicación móvil en Google play store | 8 |
| 2.1 Firmar la Aplicación Móvil | 9 |
| 2.2 Iniciar Sesión en Google Play Console | 9 |
| 2.3 Sección Ficha del Play Store | 9 |
| 2.4 Sección de versiones de app | 9 |
| 2.5 Sección Acceso a la aplicación | 9 |
| 2.6 Sección Anuncios | 9 |
| 2.7 Sección de clasificados de contenido | 9 |
| 2.8 Sección Audiencia Objetivo | 9 |
| 2.9 Categoría de la Aplicación | 9 |
| 2.10 Sección de Producción | 9 |
| 3. Credenciales de acceso para el Sistema Web y Aplicación | 9 |
| 4. Repositorio del código fuente del sistema web y aplicación móvil | 9 |

1. Despliegue del sistema web en Digital Ocean

A continuación, se muestra el proceso llevado a cabo para el despliegue y posteriormente producción del Sistema Web, para el sistema web se uso dos tipos de base de datos no relaciones las cuales son Firebase que se usó para la autenticación de usuarios y MongoDB se uso para todos los datos

1.1 Creación de un nuevo proyecto en Firebase

Se debe crear un nuevo proyecto en Firebase y asignarle un nombre como muestra la Figura 1.

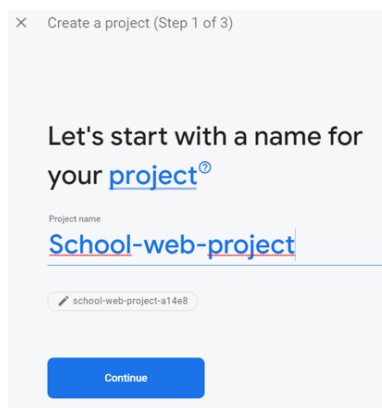


Fig 1 Crear un proyecto en Firebase

1.2 Conexión de Firebase con Angular

Lo primero será agregar una nueva aplicación de tipo web al proyecto de Firebase y así poder obtener las credenciales de acceso Figura.2

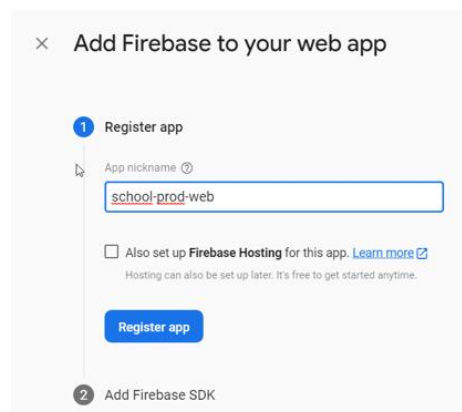


Fig 2 Conexión Firebase con Angular

Una vez registrada la aplicación, con las credenciales se procede a configurar las variables de entorno dentro del proyecto de Angular en el archivo `environments.ts`, Figura.3.

```
export const environment = {
  apiUrl: 'http://157.245.116.195:8888/api',
  apiKeyAdmin: '...',
  production: false,
  // For Firebase JS SDK v7.20.0 and later, measurementId is
  firebaseConfig: {
    apiKey: '...',
    authDomain: 'school-web-b8fc2',
    projectId: 'school-web-b8fc2',
    storageBucket: '...',
    messagingSenderId: '...',
    appId: '...',
    measurementId: '...',
  },
  partialOptions: ['Parcial 1', 'Parcial 2', 'Parcial 3'],
  topicNotification: ['Asistencia', 'General', 'Noticias'],
  quimestreoption: ['Primer Quimestre', 'Segundo Quimestre'],
  //credentials for drivers firebase
  driverFirebaseApiKey: 'AIzaSyAtp_1k7RgfaTiPwPA6fQVxEUnJ2-
```

Fig 3 Configuración de credenciales en Angular.

1.3 Creación de la base de datos en MongoDB

Se ingresa a mongo desde la consola Figura 4.

```
root@ubuntu-school-app: /var/www
root@ubuntu-school-app: /var/www#
root@ubuntu-school-app: /var/www# mongo 157.245.116.195 --port 27018 -u admin -p --authenticationDatabase admin
```

Fig 4 Ingreso a mongo desde consola

Se crea la base de datos en mongo desde la consola, Figura.5.

```
>
> show dbs
READ_ME_TO_RECOVER_YOUR_DATA 0.000GB
admin                        0.000GB
config                      0.000GB
local                       0.000GB
school-prod                 0.001GB
>
>
> use school-test
switched to db[school-test]
>
```

Fig 5 Creación de base de datos dese la consola

Se crea usuarios en la base desde la consola, Figura 6.

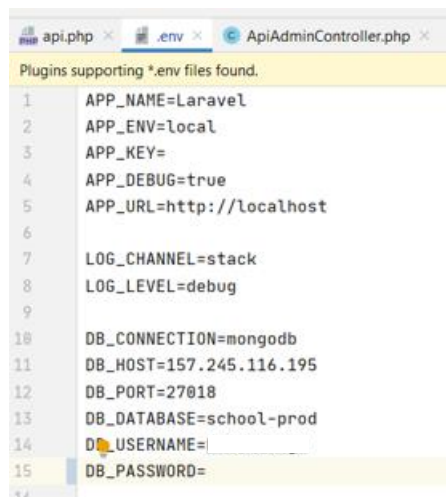
```

> show dbs
READ_ME_TO_RECOVER_YOUR_DATA 0.000GB
admin 0.000GB
config 0.000GB
local 0.000GB
school-prod 0.001GB
>
>
> use school-test
switched to db school-test
> db.createUser({
...   user:"admin",
...   pwd:"password1234",
...   roles:[
...     {
...       role:"readWriteAnyDatabase",
...       db:"admin"
...     }
...   ],
...   mechanisms:[
...     "SCRAM-SHA-1"
...   ]
... })

```

Fig 6 Usuarios en la base de datos

Se crea la conexión con angular con sus respectivas credenciales, Figura 7.



```

1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mongodb
11 DB_HOST=157.245.116.195
12 DB_PORT=27018
13 DB_DATABASE=school-prod
14 DB_USERNAME=
15 DB_PASSWORD=
16

```

Fig 7 Credenciales MongoDB

Se realiza la conexión a mongo usando robo3t, Figura 8.

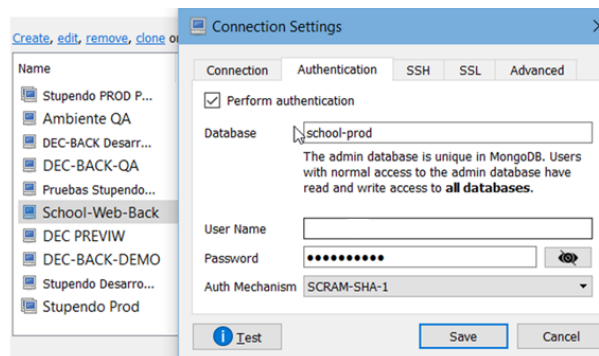


Fig 8 Credenciales en Robo3T

1.4 Despliegue del Front-end

Primero es necesario clonar el repositorio en el ambiente local en nuestro caso lo realizamos en un servidor Ubuntu 20.0 hospedado en Digital Ocean, el mismo contiene el proyecto back-end levantados mediante el servidor web Nginx, Figura.9.

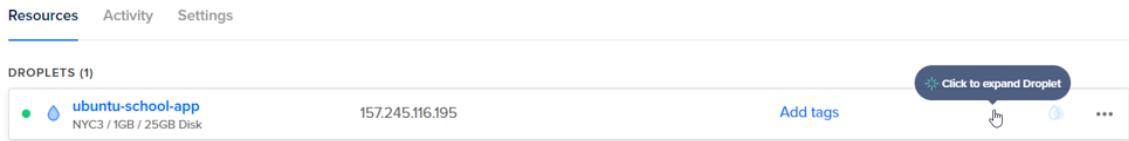


Fig 9 Servidor Ubuntu

Se debe clonar el repositorio. Figura 10.

```
root@ubuntu-school-app:/var/www/School-web-back# cd /var/www
root@ubuntu-school-app:/var/www# ll
total 16
drwxr-xr-x  4 root root 4096 Mar  1 04:29 ./
drwxr-xr-x 14 root root 4096 Mar  1 04:20 ../
drwxrwxrwx  5 root root 4096 Aug 16 01:18 School-web-back/
drwxr-xr-x  3 root root 4096 Mar  1 04:56 html/
root@ubuntu-school-app:/var/www# docker ps
CONTAINER ID   IMAGE                                COMMAND
```

Fig 10 Clonar Repositorio

Posteriormente creamos las carpetas para el front la cual es HTML. Figura.11.

```
drwxr-xr-x  3 root root 4096 Mar  1 04:56 html/
root@ubuntu-school-app:/var/www# docker ps
CONTAINER ID   IMAGE                                COMMAND
```

Fig 11 Crear la carpeta HTML

Dirigirse a la ruta donde se tiene el Nginx. Figura.12.

```
root@ubuntu-school-app:/var/www# cd /etc/nginx/sites-available/
root@ubuntu-school-app:/etc/nginx/sites-available# ll
total 12
drwxr-xr-x  2 root root 4096 Sep  1 06:06 ./
drwxr-xr-x  8 root root 4096 May 27 06:35 ../
-rw-r--r--  1 root root 2447 Sep  1 06:06 default
root@ubuntu-school-app:/etc/nginx/sites-available#
```

Fig 12 Ruta nginx

Se debe editar el archivo default donde se configura la Url del servidor y la ruta del HTML, Figura.13.

```

# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal9.
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name 157.245.116.195;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ /index.html;
    }

    # pass PHP scripts to FastCGI server
    #
}

```

Fig 13 Configurar la Url del servidor y la ruta del HTML

Por ultimo se debe reiniciar el servidor nginx, Figura.14.

```

root@ubuntu-school-app:/etc/nginx/sites-available# nano default
root@ubuntu-school-app:/etc/nginx/sites-available# sudo service nginx restart

```

Fig 14 Reiniciar el servicio nginx

1.5 Despliegue del Back-end

Se debe clonar el repositorio del back, Figura 14.

```

root@ubuntu-school-app:/var/www
root@ubuntu-school-app:/var/www# ll
total 16
drwxr-xr-x  4 root root 4096 Sep  1 06:01 ./
drwxr-xr-x 14 root root 4096 Mar  1 2021 ../
drwxrwxrwx  6 root root 4096 Sep  1 07:08 School-web-back/
drwxr-xr-x  3 root root 4096 Sep  1 06:42 html/

```

Fig 15 Clonar repositorio del back

Se debe construir el docker. Figura.15.

```

root@ubuntu-school-app:/var/www# cd School-web-back/
root@ubuntu-school-app:/var/www/School-web-back# docker-compose build

```

Fig 16 Construcción del docker

Posteriormente se debe levantar el docker. Figura.16.

```

root@ubuntu-school-app:/var/www# cd School-web-back/
root@ubuntu-school-app:/var/www/School-web-back# docker-compose up -d

```

Fig 17 Levantar el docker

Se verifica que se levantó el docker correctamente. Figura.18.

```

root@ubuntu-school-app:/var/www/School-web-back# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
4bf1025afe     nginx:stable-alpine                "/docker-entrypoint..." 6 months ago  Up 5 minutes  0.0.0.0:8888->80/tcp, :::8888->80/tcp  nginx
f6a3ceac24     mysql:5.7                          "docker-entrypoint.s..." 6 months ago  Up 7 minutes  33060/tcp, 0.0.0.0:33069->3306/tcp, :::33069->3306/tcp  mysql
c29753075bd     school-web-back_php                "docker-php-entrypoi..." 6 months ago  Up 5 minutes  9000/tcp, 0.0.0.0:9000->80/tcp, :::9000->80/tcp  php

```

Fig 18 Verificar levantamiento del docker

Dentro del docker se debe ejecutar los siguientes comandos. Figura.19.

```

root@ubuntu-school-app:/var/www/School-web-back# docker exec -it 4c29753075bd /bin/sh
/var/www/html # php artisan config:cache
Configuration cache cleared!
Configuration cached successfully!
/var/www/html # php artisan cache:clear
Application cache cleared!
/var/www/html # php artisan view:clear
Compiled views cleared!

```

Fig 19 Comandos corridos en el docker

Se verifica que el proyecto se levanto en el docker correctamente. Figura.20.

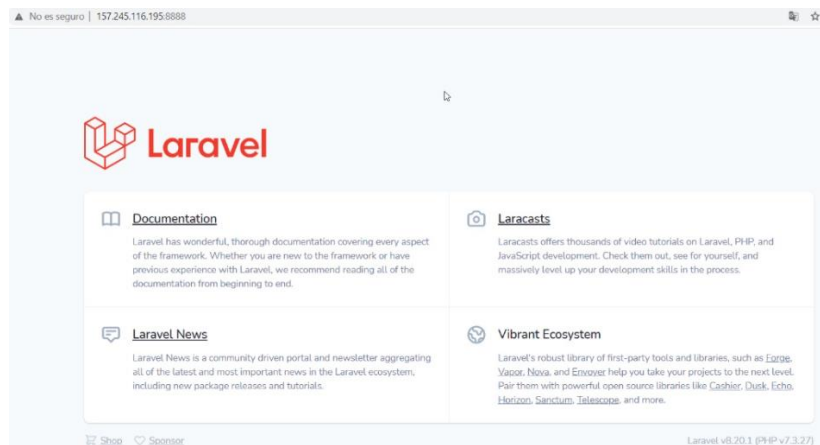


Fig 20 Proyecto levantado correctamente

2. Despliegue de la aplicación móvil en Google play store

- 2.1 Firmar la Aplicación Móvil
- 2.2 Iniciar Sesión en Google Play Console
- 2.3 Sección Ficha del Play Store
- 2.4 Sección de versiones de app
- 2.5 Sección Acceso a la aplicación
- 2.6 Sección Anuncios
- 2.7 Sección de clasificados de contenido
- 2.8 Sección Audiencia Objetivo
- 2.9 Categoría de la Aplicación
- 2.10 Sección de Producción

3. Credenciales de acceso para el Sistema Web y Aplicación

Para acceder al Sistema Web en producción, ingresar a la siguiente URL:

<http://157.245.116.195/info-school>

Credenciales para el perfil Administrador:

- Usuario: admin
- Correo de Usuario: schoolwebdescubrir001@gmail.com
- Contraseñas: Generica1 @

Para acceder a la Aplicación Móvil, se encuentra en la Play Store de Google.

4. Repositorio del código fuente del sistema web y aplicación móvil

El código fuente y la documentación del todo el proyecto se encuentran en las siguientes Urls:

- Sistema Web

Front-end

<https://github.com/Bryan-Farinango/School-web>

Back-end

<https://github.com/Bryan-Farinango/School-web-back>

- Aplicación Móvil

<https://github.com/Bryan-Farinango/school-mobil>

- Documentación

https://github.com/JosselynVela/Documentacion_Tesis