“Plan Libélula”
Versión 1.0

Documento: Manual de Arquitectura

Autores:
Marco Mancha
Ramón Romero
Iancarlo Rosas
Mariana Pérez
Carlos Román
Francisco Monroy

27 de Abril del 2018

Contenido

1. Introducción

El objetivo de este manual es brindar un documento de consulta el cual facilite el desarrollo y mantenimiento del proyecto plan libélula. En este documento se encontrarán todos los artefactos utilizados en la fase de diseño y desarrollo, para que cualquier persona pueda agregar funcionalidad o darle mantenimiento al sistema.

2. Descripción general

Debido a que Plan Libélula Desarrollo Integral carecía de un sistema que pudiera; gestionar la liga, tener el registro de los equipos que están afiliados, facilitar seguimiento a las jugadoras y poder consultar los resultados de los partidos. Y es por esto que CaveLabs desarrolló una solución para mejorar esta situación.

La solución presentada consta de distintas secciones para gestionar todos los elementos inherentes al desarrollo de la liga en cuestiones administrativas. Primero se propone un gestor de todos los participantes de la liga, como lo son equipos y jugadoras que los componen. Esta sección permite registrar y validar de forma eficiente a las jugadoras, coaches y equipos que formarán parte de un torneo en específico.

Posteriormente se propone un gestor de liga en el cual el administrador será capaz de crear y administrar una liga de fútbol. Esta sección permite gestionar el calendario del torneo, generando de forma automática el rol de juegos y posteriormente permite llevar el control de los partidos alimentando así las estadísticas, las cuales incluyen los resultados, goles, tarjetas y notas relevantes de cada partido.

Finalmente, el departamento de CaveLabs propone desarrollar una página web en donde cualquier persona interesada en la liga podrá ser capaz de ver información relevante al desarrollo de la liga incluyendo las estadísticas de los torneos, resultados de los partidos e información tanto de los equipos como de las jugadoras. De igual forma podrán acceder a las convocatorias para próximos torneos y se les facilitará una forma para ponerse en contacto con la organización para poder realizar su pre-registro a una torneo próximo.

3. Arquitectura General

3.1. Flujo Lógico

El diagrama se encuentra en la siguiente liga:
https://drive.google.com/open?id=1n5zSDf7m37ailxkm_a2khBBhjKEHoMl8

3.2. Diagrama despliegue

El diagrama se encuentra en la siguiente liga:
https://drive.google.com/open?id=184WZwEthLFx_zTLkTEk9sghTRyKIiPpo

3.3. Diseño base de datos

El diagrama se encuentra en la siguiente liga:
https://drive.google.com/open?id=11Dn-F70oRFRt4i01gb0xLuiUNvoxZOXy

3.4. Diagrama de vistas y componentes

El diagrama se encuentra en la siguiente liga:
https://drive.google.com/open?id=1k39jX4_RsPMYxqQb-mshMglnBmSAp4IA

4. Hardware/Software

Las siguientes versiones de herramientas y equipo, fueron utilizadas para el desarrollo del sistema, se desea que si se llega a desarrollar una extensión del mismo se utilicen las mismas herramientas.

Herramientas

- Github: la línea base del proyecta se encuentra en https://github.com/CaveLabs-1/Libelulas
- Atom 1.24.0
- Pycharm: 2017.3.3

- Git Kraken: 3.3.4
- Git 2.7.4

Navegadores

- Chrome: 64.0.3282.186+
- Opera: 50.0.2762.67+
- Safari: 11.0.3+
- Firefox: 58.0.2+

Sistemas operativos

- MacOS: High Sierra 10.1.7+
- Ubuntu: 16.04 LTS+
- Windows: Windows 10 1709+

Lenguajes

- Python 3.6

Framework Back end

- Django 2.0.2

Herramientas Front end

- Bootstrap: v4.0.0
- NodeJs: 8.9.4 LTS
- Materialize: 1.0.0

Gestor de Base de datos

- PostgreSQL: 10.2

5. Dependencias del sistema

La siguiente lista de paquetes y módulos, son necesarios para el desarrollo del sistema.

- cairocffi==0.8.0
- CairoSVG==2.1.3
- cffi==1.11.5
- cssselect2==0.2.1
- defusedxml==0.5.0
- Django==2.0.2
- html5lib==1.0.1
- pdfrw==0.4
- Pillow==5.0.0
- pkg-resources==0.0.0
- psycopg2==2.7.4
- psycopg2-binary==2.7.4
- pycparser==2.18
- Pyphen==0.9.4
- pytz==2018.3
- reportlab==3.4.0
- six==1.11.0
- tinycss2==0.6.1
- WeasyPrint==0.42.3
- webencodings==0.5.1
- freezegun==0.3.10

6. Especificaciones técnicas

Hosting:

Se contrató un droplet con las siguientes características:

MEMORY	VCPUS	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	$5/mo $0.007/hr

El servidor está configurado con los siguientes servicios:

- Ubuntu 16.04
- PostgreSQL 10.2
- Django 2.0.2
- Nginx
- Gunicorn
- Python 3.6

Dominio:

El domino que se está utilizando es el de planlibelula.com y el proveedor de servicios es GoDaddy.

7. Pruebas de unidad e integración

Se utilizó la suite de testing de Django de Unit Testing. Cada prueba se diseñó a partir de los casos de aceptación definidos previamente. Para un mayor entendimiento de cómo construir y ejecutar pruebas en Django consultar el siguiente link:

https://docs.djangoproject.com/en/2.0/topics/testing/overview/

8. Configuración servidor

La configuración del servidor se hizo utilizando Nginx, Ubuntu 16.04 y PostgreSQL. La guía paso a paso para poder configurar el servidor se encuentra en el siguiente link:

https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu-16-04

9. Actualización servidor

En la raíz del proyecto existe un script que sirve para poder activar el virtual environment, bajar las actualizaciones del repositorio, realizar las migraciones necesarias y actualizar el servidor en caso de que haya algún cambio en el proyecto. El nombre del script es *deploy.sh* por lo que si es necesario actualizar el servidor son necesarios los siguientes pasos:

- Pull request a la rama *master.*
- Actualizar el archivo *requirements.txt* en caso de que se haya instalado alguna otra dependencia.
- Actualizar el archivo *deploy.sh* en caso de que se haya creado alguna otra app de la aplicación Libélulas.
- Ejecutar el script con el siguiente comando: *source deploy.sh*

Con este último comando, el repositorio del proyecto, las dependencias y migraciones quedarán actualizadas y el intermediario entre el servidor y la aplicación web también será actualizado.

10. Configuración dominio

En la siguiente liga se encuentra un tutorial paso a paso para poder configurar el dominio en el portal de Digital Ocean en caso de que el dominio llegara a cambiar:

https://github.com/CaveLabs-1/Wiki/blob/master/Arquitectura/Manuales/Manual%20Arquitectura%20General.md#dominio

11. Habilitar HTTPS: Certificado SSL

En la siguiente liga se encuentra un tutorial paso a paso para poder configurar el certificado de SSL y habilitar el HTTPS por medio de una sesión por SSH.

https://github.com/CaveLabs-1/Wiki/blob/master/Arquitectura/Manuales/Manual%20Arquitectura%20General.md#seguridad