

“CADHU”
Versión 1.0

Documento: Manual de Arquitectura

Autores: Alejandro Salmón Félix Díaz
Filiberto Francisco Vázquez Rodríguez
Santiago Valencia Corona
Luis Rodriguez Santos
Marco Antonio Luna Calvillo
Mauricio Hernandez Siles

30 de abril de 2018

Contenido

Introducción	4
Descripción general	4
Arquitectura General	
Flujo Lógico	4
Diagrama despliegue	4
Diagrama de vistas y componentes	4
Diseño base de datos	4
Hardware/Software	4
Dependencias del sistema	4
Especificaciones técnicas	4
Pruebas de unidad e integración	4
Configuración servidor	4
Actualización servidor	4
Configuración dominio	4
Habilitar HTTPS: Certificado SSL	4
Glosario	4

1. Introducción

El objetivo de este manual es brindar un documento de consulta el cual facilite el desarrollo y mantenimiento del proyecto CADHU. 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

La problemática presentada por *CADHU* consiste principalmente en el uso de distintas herramientas. Las bases de datos utilizadas para poder realizar un proceso administrativo no están concentradas en un solo sistema, sino que los ejecutivos tienen que acudir a distintas aplicaciones web para poder acceder a ciertas bases de datos. Las áreas con deficiencia incluyen prospectos, clientes, cursos y registro de pagos.

El área de mayor oportunidad del proyecto se encuentra en la fase de captación de prospectos como clientes donde la desorganización produce una caída de interés. La causa consiste en que al contactar a los prospectos, se hace varias veces por error y se crea un conflicto en el registro informal de la información al no tener una manera normalizada para almacenar dicha información. Por otro lado, las bases de datos de los eventos están fragmentadas lo que causa retrabajo el momento de registrar sus clientes y tener seguimientos del registro de pagos. Como consecuencia, esto constituye a un problema de logística e ineficiencia, debido a la falta de unicidad de datos, la información no puede ser manipulada de manera que se logre generar reportes para poder tomar decisiones empresariales y generar un valor agregado a la información. Finalmente, se atribuye a fallas dentro del sistema de contabilidad donde se pierde un control preciso de los ingresos y gastos que se generan.

La manera en la que se quieren resolver las problemáticas que enfrenta actualmente CADHU con sus sistemas existentes es agregando un sistema que consta principalmente de los módulos de prospecto, clientes, cursos y pagos. El módulo de gestor de prospectos se encargaría de registrar inicialmente la información de los prospectos para que una vez notado su nivel de interés, se le invite a algún curso y pueda después realizar su pago creando un cliente para el grupo en específico. La información quedará registrada en el perfil de cada prospecto.

3. Arquitectura General

3.1. [Flujo Lógico](#)

3.2. [Diagrama despliegue](#)

3.3. Diagrama de vistas y componentes

3.3.1. [Diagrama de vistas](#)

3.3.2. [Guia de estilos](#)

3.4. Diseño base de datos

3.4.1. [MER](#)

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 proyecto se encuentra en <https://github.com/CaveLabs-1/CADHU>
- 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

- Materialize: 1.0.0
- JQuery Confirm
- Data Tables

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.

- Django==2.0.2
- Pillow==5.0.0
- PyYAML==3.12
- django-widget-tweaks==1.4.2
- jdcal==1.3
- et-xmlfile==1.0.1
- odfpy==1.3.6
- openpyxl==2.5.1
- pip==10.0.0
- psycpg2==2.7.4
- psycpg2-binary==2.7.4
- pycparser==2.18
- Pyphen==0.9.4
- pytz==2018.3
- setuptools==28.8.0
- tablib==0.12.1
- unicodecsv==0.14.1
- xlrd==1.1.0
- xlwt==1.3.0

6. Especificaciones técnicas

MEMORY	VCPUS	SSD DISK	TRANSFER	PRICE
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr

El servidor está configurado con los siguientes servicios:

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

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

Después de iniciar sesión se insertan los siguientes comandos.

- `sudo apt-get update`
- `sudo apt-get upgrade`

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>