



Ingeniería Electrónica
Proyectos Computacionales Aplicados a Ingeniería Electrónica
Ing. José Anibal Silva de Los Angeles
Aux. Fernando Mardoqueo Paxtor Sam

CARNÉ:	201903870	FECHA:	12/05/2022
NOMBRE:	Andy Gudiel Perez Molina		
REGISTRO ADMIN.	2466336701014		

CARNÉ:	201903890	FECHA:	12/05/2022
NOMBRE:	Henry Eduardo Ramírez Rojas		
REGISTRO ADMIN.	3050059950117		

Proyecto Refugio (Manual Técnico)

Git de Andy Perez:

<https://github.com/AndyPerezMolina/Proyecto-refugio.git>

Git de Henry Ramírez:

<https://github.com/EduardRam2001/Proyecto-Refugio.git>

INTRODUCCIÓN

El manual detalla los aspectos técnicos e informáticos del programa “Refugio” con la finalidad de explicar la estructura del aplicativo al personal que quiera administrarlo, editarlo o configurarlo. La siguiente guía se encuentra dividida en las herramientas que se usaron para la creación del software con una breve explicación paso a paso.

Este manual tiene como finalidad s instruir a la persona que quiera administrar, editar o configurar la página web usando las debidas herramientas.

OBJETIVO

Dar a conocer el uso adecuado de la página web en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades que conforman el buen funcionamiento de la página.

HERRAMIENTAS UTILIZADAS EN EL DESARROLLO

1. Visual Studio Code

Es un editor de texto el cual es caracterizado por sus amplias funcionalidades a la hora de desarrollar software, tiene incorporado la estructura en la sintaxis de diferentes lenguajes de programación con una interfaz amigable, en el desarrollo de la página web “Refugio” se trabajó con esta herramienta de marcado ya que indica en donde está ubicado los errores de sintaxis y la codificación se maneja de forma dinámica teniendo en cuenta que se requieren librería para un uso en específico.

2. GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, Manual técnico del sistema dentro de un proyecto. Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls

3. Docker

Envuelven una pieza de software en un sistema de archivos completo que contiene todo lo necesario para funcionar: código, tiempo de ejecución, herramientas del sistema, las bibliotecas del sistema - cualquier cosa que se puede instalar en un servidor. Esto garantiza que el software se ejecutará siempre la misma, independientemente de su medio ambiente. (Docker, s.f.) Se usa Docker para poder desarrollar de una manera más cómoda, ya que este software contiene millones de librerías y herramientas para poder codificar.

INSTALACION DE APLICACIONES

Instalación de Docker

Requisitos minimo:

- Windows 10 64-bit: Pro, Enterprise, or Education (Build 16299 o superior).
- Procesador de 64 bits.
- 4 GB RAM.
- Habilitar en la BIOS la característica (el nombre varia en función del procesador que tenga tu equipo) “Intel VR” o “VR-x” o “Virtualization Technology” o «AMD-V».

El programa se descarga en el siguiente enlace en la página oficial de Docker

<https://www.docker.com/get-started/>

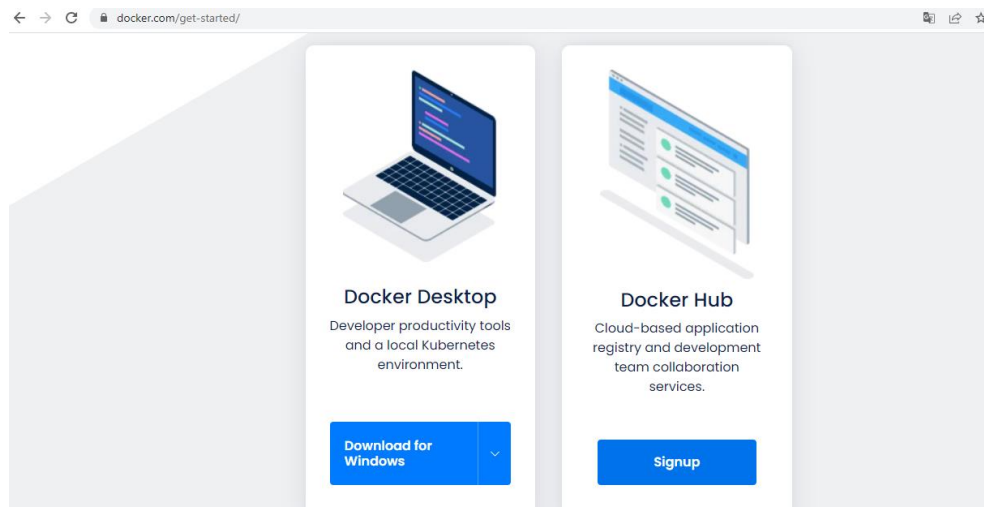


Figure 1 Instalación de Docker

Instalación de WSL2

Requisitos minimo:

- Para sistemas x64 se requiere como mínimo la versión 1903 con build 18362 o superior.
- Para sistemas ARM64 se requiere como mínimo la versión 2004 con build 19041 o superior.

Para instalar WSL2 es por medio de comando, en el siguiente enlace esta todos los pasos a seguir:

<https://docs.microsoft.com/es-es/windows/wsl/install>

LENGUAJE DE PROGRAMACIÓN

Pyhton

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

DICCIONARIO DE DATOS

Diccionario de datos adopción Persona

Adopcion_Persona			
Nombre de campo	Tipo de dato	Tamaño	Descripcion
nombre	CharField	50	Nombre del usuario
apellidos	CharField	70	Apellido del usuario
edad	IntegerField(Edad del usuario
telefono	CharField	8	Numero de telefono del usuario
email	EmailField		E-mail del usuario
direccion	TextField		Dirreccion del usuario

Diccionario de datos adopción solicitud

Adopcion_solicitud			
Nombre de campo	Tipo de dato	Tamaño	Descripcion
numero_mascota	IntegerField		Numero de mascota de adopcion
razones	TextField		Razones para la adopcion
persona_id	ForeignKey		Idetificador de persona

Diccionario de datos de ingresar una mascota

mascota_mascota			
Nombre de campo	Tipo de dato	Tamaño	Descripcion
nombre	CharField	50	Nombre del animal
raza	CharField	50	Raza del animal
sexo	CharField	10	Sexo del animal
edad_aproximada	IntegerField		Edad del animal
fecha_rescate	DateField		Fecha de rescate del animal
alimentacion	CharField	100	Alimentacion del animal
enfermedades	CharField	100	Enfermedad que posee el animal
persona	ForeignKey		Persona que lo va adoptar
vacuna	ManyToManyField		Vacuna que posee el animal

Diccionario de datos Relacion mascota y vacuna

mascota_mascota			
Nombre de campo	Tipo de dato	Tamaño	Descripcion
mascota_id	IntegerField		Numero de mascota en el registro
vacuna_id	IntegerField		Numero de vacuna que posee el animal

Diccionario de datos vacuna

mascota_vacuna			
Nombre de campo	Tipo de dato	Tamaño	Descripcion
nombre	CharField	50	Nombre de la vacuna

CONFIGURACION EN DJANGO

Configuración de horario y lenguaje

```
LANGUAGE_CODE = 'es-mx'
TIME_ZONE = 'America/Guatemala'
USE_I18N = True
USE_TZ = True
```

Configuración de correo para el reset password

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
EMAIL_USE_TLS = True
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_HOST_USER = 'proyecto980ingenieria@gmail.com'
EMAIL_HOST_PASSWORD = 'Clase_980'
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
```

Configuración para determinar el mínimo de longitud de la contraseña

```
UTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
        'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
        'OPTIONS': { 'min_length': 12,}
    }
]
```

Configuración de la base de datos:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': os.environ.get('DB_NAME'),  
        'USER': os.environ.get('DB_USER'),  
        'PASSWORD': os.environ.get('DB_PASSWORD'),  
        'HOST': os.environ.get('DB_HOST'),  
    }  
}
```

COPIA DEL CODIGO CON SUS COMPLEMENTOS

Para descargar el código completo, puede usar cualquiera de los dos links, que se presentan:

Git de Andy Perez:

<https://github.com/AndyPerezMolina/Proyecto-refugio.git>

Git de Henry Ramírez:

<https://github.com/EduardRam2001/Proyecto-Refugio.git>