



# Analíticas Automáticas de Datos Masivos en una Plataforma Digital Educativa

Ángel Perea Arias  
Ingeniería en Sistemas Audiovisuales y Multimedia





# Índice

1. Introducción
2. Objetivos
3. Estado inicial
4. Prototipo con MongoDB y Matplotlib
5. Prototipo con Elasticsearch y Kibana
6. Conclusiones

---

# INTRODUCCIÓN



## Plataformas digitales

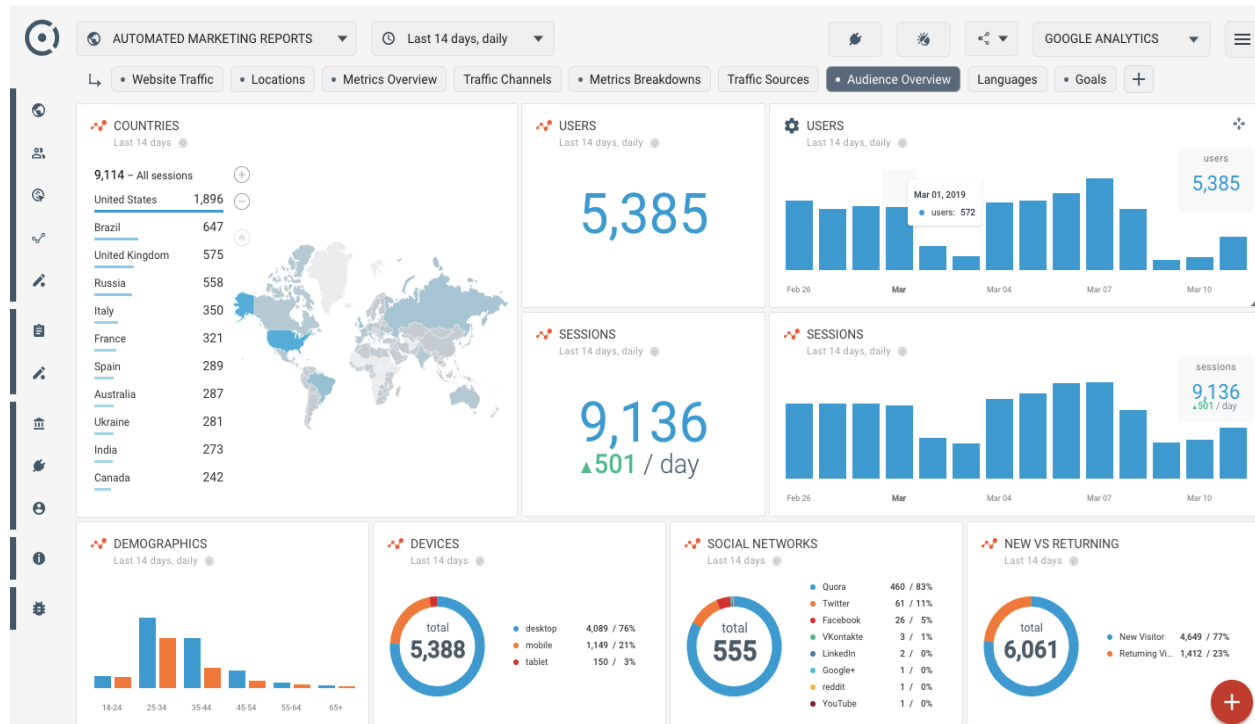
- Aumenta la oferta y la demanda
- Entorno muy competitivo
- Avances de las tecnologías

## Importancia de conocer a los usuarios

- Aumentar la satisfacción
- Aumentar la retención
- Detectar tendencias de uso

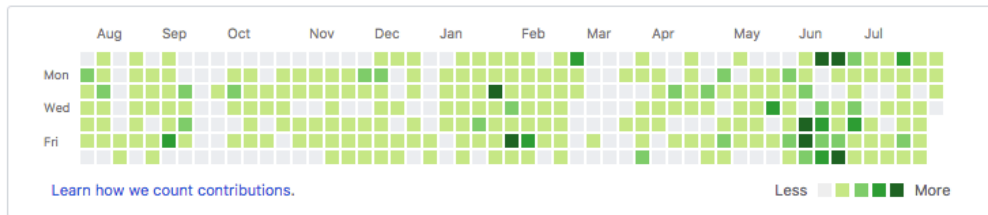


# Google Analytics



# Estadísticas GitHub

1,548 contributions in the last year



elastic/elasticsearch

Watch 2.8k Star 50k Fork 17.2k

Code Issues 2.5k Pull requests 348 Actions Projects 1 Security Insights

Pulse

Contributors

Community

Commits

Code frequency

Dependency graph

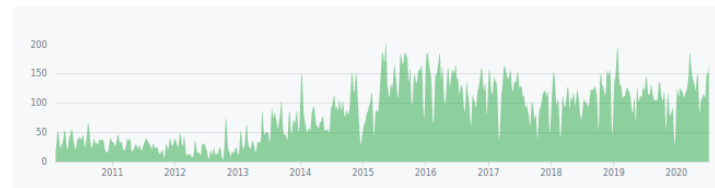
Network

Forks

Feb 7, 2010 – Jul 17, 2020

Contributions: Commits ▾

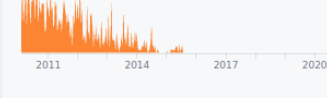
Contributions to master, excluding merge commits



kimchy

4,738 commits 1,401,537 ++ 1,064,923 --

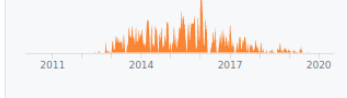
#1



s1monw

2,880 commits 1,330,070 ++ 1,316,286 --

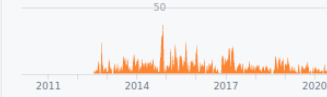
#2



martijnvg

2,568 commits 411,398 ++ 277,474 --

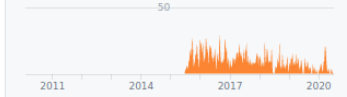
#3



jasontedor

2,565 commits 383,687 ++ 299,085 --

#4



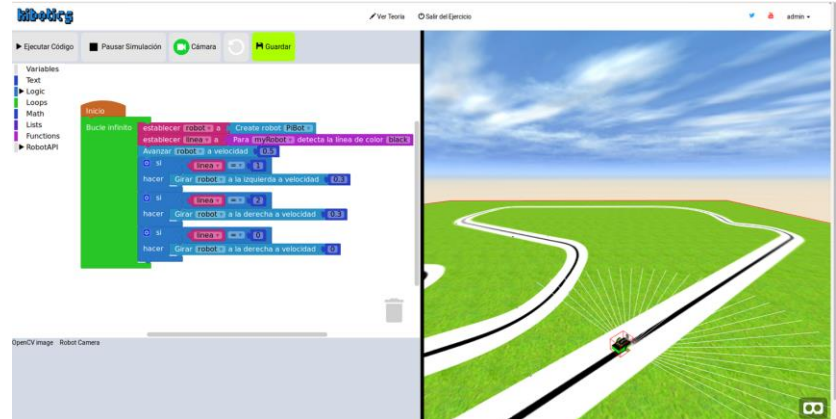
# Plataforma educativa Kibotics

- Orientado a niñ@s y adolescentes
- Ejercicios online robótica y visión artificial
- Virtualización y simulación de robots
- Programación en Scratch o Python

SCRATCH




kibotics



---

# OBJETIVOS





Dotar a la plataforma Kibotics con una herramienta de visualizaciones web automática para monitorizar a sus usuarios

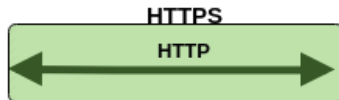


---

**ESTADO INICIAL**



CLIENTE



WebSocket



SERVIDOR

APACHE

WSGI

DJANGO

Controlador

Vistas

Modelo

urls.py

views.py

models.py

Daphne  
WebSocket  
Server

***SIMULADOR  
WEB-SIM***

Base de Datos



Ficheros de Log





## Ficheros de log

- Log in: "1 | date | user name | user IP | HTTP\_USER\_AGENT"
- Log out: "2 | date | user name | user IP | HTTP\_USER\_AGENT"
- Comienzo ejercicio: "3 | date | user name | user IP | simulation type | exercise ID | host IP | HTTP\_USER\_AGENT"
- Fin ejercicio: "4 | date | user name | user IP | simulation type | exercise ID | host IP | HTTP\_USER\_AGENT"
- Error 500: "5 | 500 Internal Server Error"

---

# PROTOTIPO CON MONGODB Y MATPLOTLIB



CAPTURA

ALMACENAMIENTO

VISUALIZACIÓN

*matplotlib*



mongoDB



# Almacenamiento de información

## Librería Pymongo

Los documentos almacenados siguen la misma estructura de campos que en los ficheros de Log

## 5 Colecciones MongoDB

- newSession
- endSession
- newSimulation
- endSimulation
- applicationError

```
user_agent = request.META['HTTP_USER_AGENT']
mydict = {
    "date" : str(datetime.now().strftime("%d/%m/%Y %H:%M:%S")),
    "username" : user.username,
    "client_ip" : client_ip,
    "simulation_type" : simulation_type,
    "exercise_id" : exercise_id,
    "host_ip" : host.ip,
    "container_id" : "LOCAL CONTAINER",
    "user_agent" : "user_agent"
}
mydb["newSimulation"].insert_one(mydict)
```

# Generación de visualizaciones

# Librería Matplotlib

1. Extracción de datos (Pymongo)
2. Procesado de datos (Python)
3. Creación de la visualización
4. Ajuste diseño gráfico
5. Renderización
6. Envío al cliente (Plantillas Django)

[illegible]

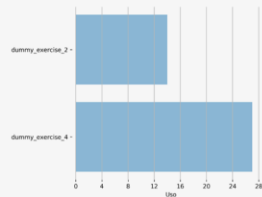




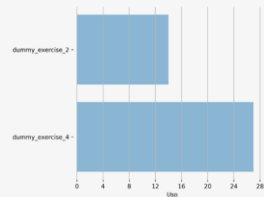
# Versión integrada

TIEMPO EN SIMULACIONES[Minutos]

TOTAL



MEDIO



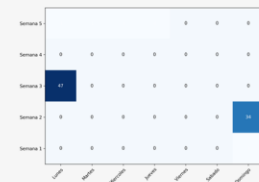
MAPA DE USUARIO



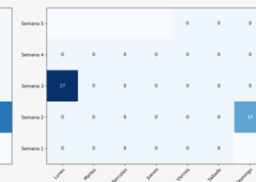
## ANÁLÍTICAS DE GRUPO 'admin' PARA 30 DIAS

MAPAS DE ACTIVIDAD [Minutos]

Mapa calor sesiones



Mapa calor simulaciones



INICIOS POR DIA DE LA SEMANA

Sesiones

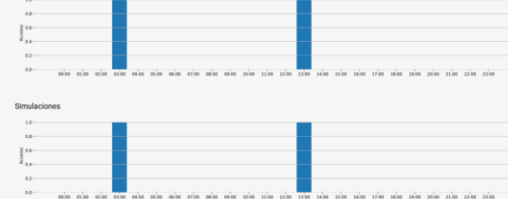


Simulaciones

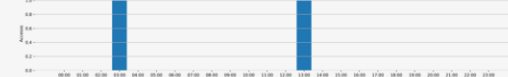


INICIOS POR HORA DEL DIA

Sesiones



Simulaciones





## Inconvenientes

- El cruce de los datos ralentiza el procesamiento
- La renderización de las imágenes es lenta
- Visualizaciones no interactivas
- Falta información de los visitantes de la aplicación

---

# PROTOTIPO CON ELASTICSEARCH Y KIBANA





# Almacenamiento de información

## Librería Elasticsearch Python

- Nuevas sondas para visitantes
- Unificación índices para eventos similares
- Nuevos campos (Sistema operativo, dispositivo y navegador)

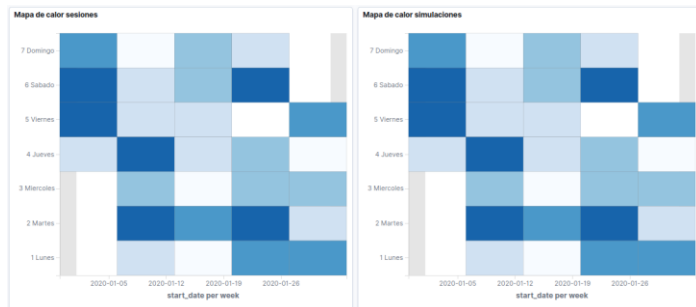
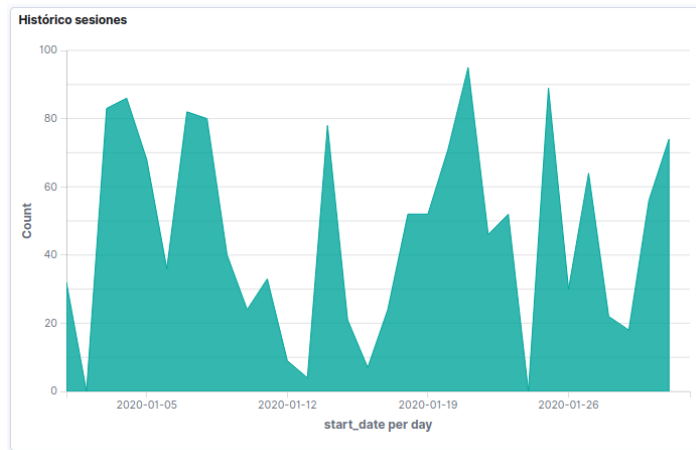
## 4 Índices Elasticsearch

- kibotics\_session\_log
- kibotics\_simulation\_log
- kibotics\_visit\_log
- kibotics\_error\_log

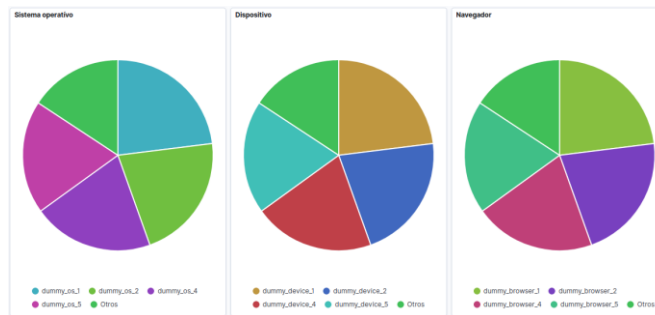
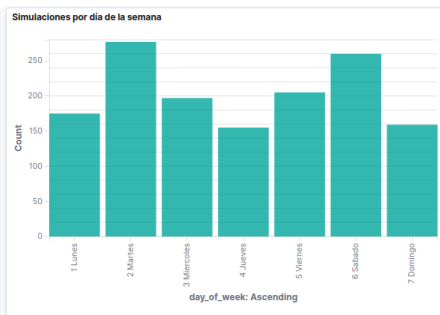
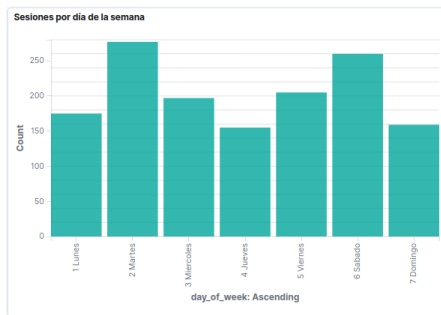
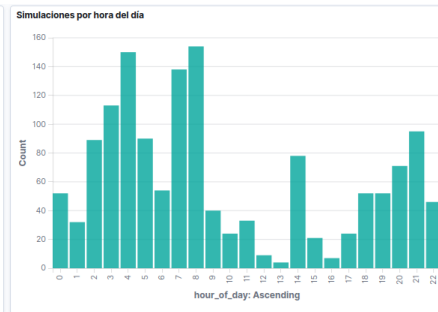
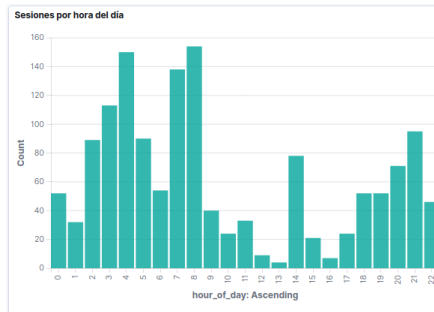
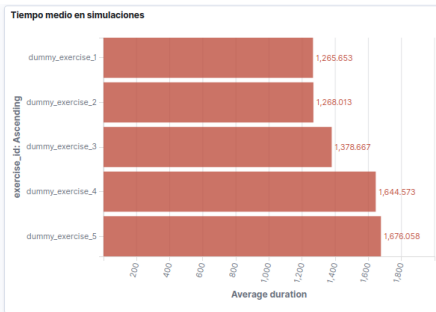
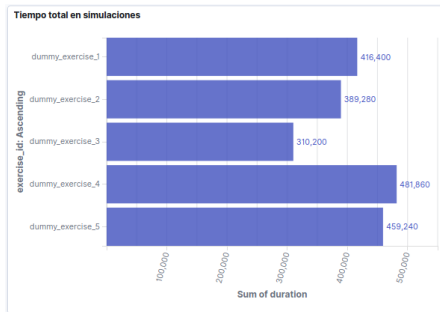
```
user_agent = parse(request.META['HTTP_USER_AGENT'])
try:
    loc = {'lat': geolite2.lookup(get_client_ip(request)).location[0],
          'lon': geolite2.lookup(get_client_ip(request)).location[1]}
except:
    loc = {'lat': 0.0, 'lon': 0.0}
VisitDocument(
    date=datetime.now(),
    client_ip=get_client_ip(request),
    browser=user_agent.browser.family + " " + user_agent.browser.version_string,
    os=user_agent.os.family + " " + user_agent.os.version_string,
    device=user_agent.device.family,
    location = loc
).save()
```

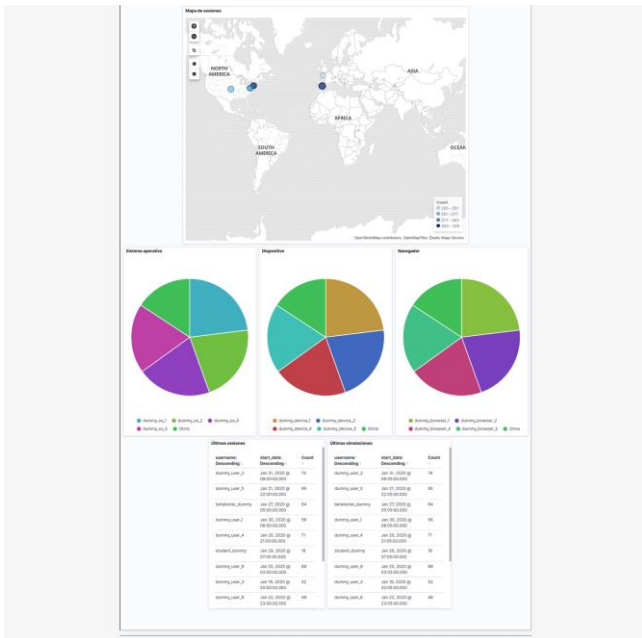


# Visualizaciones Kibana









---

# CONCLUSIONES



Objetivo cumplido, se ha desarrollado una herramienta de analíticas web automáticas:

1. Aumento de sondas para grabar eventos
2. Implementación de BBDD persistente
3. Integración en la web de visualizaciones web automáticas

Trabajos futuros:

- Monitorización de más eventos
- Detección de tendencias en el uso de la plataforma