

# POC

Esteban Quirós Alvarado, 2014012892,

graphicx float spverbatim listings

## I. DIAGRAMA DE COMPONENTES

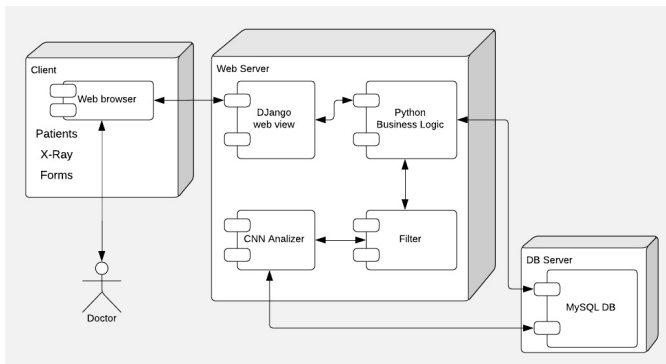


Figura 1. Diagrama de Componentes.

### I-A. Componentes

- **Doctor:** es el usuario principal del sistema y encargado de ingresar los datos de los pacientes, imágenes de los rayos-X o buscar al paciente en caso de que ya tenga historial.
- **Web browser:** al ser una aplicación web, el usuario debe utilizar un web browser y dirigirse a la dirección web correcta.
- **Django web view:** componente creado con la herramienta Django, es la parte que va a interactuar con el usuario, el front end de la aplicación.
- **Python Business Logic:** es el back end de la aplicación, debe tener comunicación con la base de datos.
- **Filter:** componente que se encarga de aplicar los filtros necesarios a las imágenes, para su posterior análisis.
- **CNN Analyzer:** una vez que a la imagen se le han aplicado los filtros necesarios, este componente procede a analizar y calcular la edad ósea del paciente.
- **MySQL DB:** componente de guardar toda la información obtenida, tanto del análisis de la imagen, como la de los datos personales del paciente.

## II. TROZOS DE CÓDIGO

### II-A. Cargar una imagen

```
import matplotlib.pyplot as plt
import numpy as np
from pylab import imread, imshow

img = imread('D:\Esteban\Desktop\
Ezatará\EZ\EZ-Images\grim.jpg')
.astype(np.float32)

image = img / 255
imshow(image)
```

- Utilización de las librerías numpy, matplotlib y pylab para cargar una imagen. Se lee la imagen con el path respectivo y luego se muestra con la función imshow que pertenece a matplotlib.

### II-B. Cargar archivo CSV

```
import numpy as np

data = np.genfromtxt('D:/Esteban/Desktop/
Ezatará/EZ-U/Aseguramiento/POC/train.csv',
delimiter = ',', skip_header=1, dtype=None)

for row in data:
    print(row)
```

- Se utiliza la librería numpy y su función de genfromtxt para leer el archivo csv, se le pasa el path, se indica el delimitador, además de indicar en el dtype como nulo, para que no exista problemas de lectura de ints, floats, booleans o strings.

### II-C. Aplicación de contraste a una imagen

```
from scipy import ndimage
import matplotlib.pyplot as plt
import numpy as np
from pylab import imread, imshow, gray, mean

img = imread('D:\Esteban\Desktop\
Ezatará\EZ\EZ-Images\grim.jpg')
.astype(np.float32)

image = mean(img, 2)

plt.imshow(image, cmap=plt.cm.gray, vmin=30,
vmax=200)

plt.axis('off')
plt.show()
```

- En este trozo se aplican varias funciones de las librerías numpy, matplotlib y pylab, se pasa la imagen por medio de su dirección, se hace un mapeo de la imagen en gris, se dicta la intensidad para el contraste para luego mostrar el resultado de las modificaciones.

### II-D. Login en Django

```
from django.views.generic.edit
import FormView
from django.http.response
import HttpResponseRedirect
from django.core.urlresolvers
import reverse_lazy
from django.contrib.auth.forms
import AuthenticationForm
from django.contrib.auth
import login

class Login(FormView):
```

```

template_name = 'login.html'
form_class = AuthenticationForm
success_url = reverse_lazy("personas:
urlpatterns
def dispatch(self, request,
*args, **kwargs):
    if request.user.is_authenticated():
        return HttpResponseRedirect(
            self.get_success_url()
        )
    else:
        return super(Login, self).
            dispatch(request,
*args, **kwargs)
def form_valid(self, form):
    login(self.request, form.
get_user())
    return super(Login, self).
form_valid(form)

```

- En este trozo se aplican varias funciones de las bibliotecas de Django. Se importa una vista genérica del tipo Formview, se importa el formulario de de autenticación de Django y se crea la clase necesaria para el funcionamiento del login.

### III. PROBLEMAS ENCONTRADOS CON LAS HERRAMIENTAS

- A uno de los integrantes del grupo se le presentó un problema con la instalación de TensorFlow, que es necesario para Keras. La solución fue utilizar Anaconda para poder instalar tanto TensorFlow como todas las demás herramientas.

### REFERENCIAS

```

###Cargar Imagen original###
import matplotlib.pyplot as plt
import numpy as np
from pylab import imread, imshow

img = imread('D:\Esteban\Desktop\Ezatara\EZ\EZ-Images\grim.jpg').astype(np.float32)
image = img / 255
imshow(image)

```

Figura 2. Código para cargar una imagen

```

#Cargar y leer CSV
import numpy as np
data = np.genfromtxt('D:\Esteban\Desktop\Ezatara\EZ-U/Aseguramiento/POC/train.csv', delimiter=',', skip_header=1, dtype=None)
for row in data:
    print(row)
    #print(row[0], row[1], row[2]) #Para cargar el csv pero no desplegar de forma de array

```

Figura 3. Código para cargar un archivo cvs

```

###Cargar la imagen y ponerle y manejar el contraste de la misma###
from scipy import ndimage
import matplotlib.pyplot as plt
import numpy as np
from pylab import imread, imshow, gray, mean

img = imread('D:\Esteban\Desktop\Ezatara\EZ\EZ-Images\grim.jpg').astype(np.float32)
image = mean(img,2)
plt.imshow(image, cmap=plt.cm.gray, vmin=30, vmax=200)
plt.axis('off')

plt.show()

```

Figura 4. Código para aplicar y manejar contraste de una imagen

localhost:8000/login/

## User Login

Username:

Password:

Figura 5. Logind de prueba en Django