Redes Neuronales Artificiales para la Detección del Trastorno Depresivo Persistente y Trastorno Depresivo Mayor.

# Artificial Neural Networks for the Detection of Persistent Depressive Disorder and Major Depressive Disorder.

Ricardo de Jesús González Morales (1), estudiante, I.T. de Tuxtla Gutiérrez. [0.ricardogonzalez.0@gmail.com.](mailto:0.ricardogonzalez.0@gmail.com.)

Rafael Antonio Bolaños Cameras (2), estudiante, I.T. de Tuxtla Gutiérrez, [animasdelmundo2@gmail.com.](mailto:animasdelmundo2@gmail.com.)

Aída Guillermina Cossío Martínez (3), I.T. de Tuxtla Gutiérrez, [acossio\_m@yahoo.com.](mailto:acossio_m@yahoo.com)

## Artículo recibido en MES DIA, AÑO; aceptado en MES DIA, AÑO.

**Resumen.**

*Este trabajo presenta el diseño de un sistema de 3 redes neuronales conectado a un aplicativo web para la detección del trastorno depresivo mayor y distímico mediante una salida probabilística derivada de cada una de las redes neuronales. El modelo utiliza 3 redes neuronales tipo feedforward. Para el proceso de aprendizaje, se emplean valores obtenidos de exámenes realizados por psicólogos y psiquiatras para diagnosticar los trastornos en un individuo mediante el Inventario Depresivo de Beck. Dicho diagnóstico será realizado mediante el llenado de un formulario en el aplicativo web, el cual pasará los datos a las redes neuronales para realizar la predicción y obtener el resultado del diagnóstico. El aplicativo, así como las redes neuronales, serán desarrolladas en el lenguaje Python.*

**Palabras clave:** Trastorno depresivo mayor, tratorno distímico, Django, Python, redes neuronales.

# Abstract.

*This work presents the design of a system of 3 neural networks connected to a web application for the detection of major depressive and dysthymic disorder through a probabilistic output derived from each of the neural networks. The model uses 3 feedforward neural networks. For the learning process, values ​​are obtained from tests carried out by psychologists and psychiatrists that are used to diagnose disorders in an individual using the Beck Depressive Inventory. Such diagnosis will be made by filling in a form in the web application, which will pass the data to the neural networks to make the prediction and obtain the result of the diagnosis. The application, as well as the neural networks, will be developed in Python the programming languaje.*

**Keywords**: Major depressive disorder, dysthymic disorder, Django, Python, neural networks.

# Introducción.

Según la Organización Mundial de la Salud (2019), el Trastorno Depresivo Mayor es un trastorno mental bastante frecuente que tiene como sintomatología, la tristeza, la perdida de interés o placer, sentimiento de culpa, falta de concentración y baja autoestima. Puede llegar a hacerse crónico y recurrente, dificultando el desempeño y calidad de vida de la persona en todos los ámbitos; en la escuela, con la familia y en las relaciones interpersonales. Si este trastorno se hace grave, puede conducir al suicidio. La depresión puede ser tratada con terapias, pero si esta llega a un nivel moderado o grave, se necesitará de medicación y psicoterapia profesional. En México, la Secretaria de Salud dice que el Trastorno Depresivo Mayor se manifiesta a través de varios síntomas, como una respuesta a largos periodos de estrés y angustia. Existen multitud de factores que están asociados a la depresión y ansiedad. Los problemas que tenemos diariamente, pueden juntarse y desencadenar algún cuadro de estos padecimientos, especialmente si tenemos antecedentes de haber padecido alguno de estos trastornos. Es difícil que un paciente pueda detectar tempranamente la depresión, regularmente puede llegar a ser confundido con la tristeza, sin embargo, es de vital importancia poder detectarla a tiempo, para que esta no se convierta en un trastorno permanente. La diferencia entre la tristeza y la depresión, es que los síntomas suelen ser más graves, persistentes, e intensos. Este trastorno tiene como tratamiento, la psicoterapia, la cual consiste en ejercicios de relajación, que de manera conjunta ayudan a disminuir los síntomas de tristeza, ansiedad y melancolía, además de que aportan calma y mayor control en situaciones de estrés (IMMS, 2019). Sin embargo, existen casos donde la psicoterapia no es suficiente y se tiene que recurrir a medicación por medio de un especialista psiquiátrico, e incluso en casos más severos se llega a internar al paciente para tenerlo en constante observación, debido al riesgo que conlleva a que la depresión pueda ser el desencadenante de algún intento de suicidio en el paciente.

Son los jóvenes de edades comprendidas entre 15 a 29 años y en su mayoría pertenecientes a instituciones escolares los que suelen desarrollar este tipo de problemas, y uno de los mayores retos a los que Chiapas se enfrenta es que no todas las instituciones poseen un psicólogo laborando dentro de ellas, además, la cantidad de especialistas en psiquiatría y psicología en nuestro estado es pequeña comparada con la población que se enfrenta a estos trastornos. Aunado a esto, cuando se habla del Trastorno Depresivo Mayor, en una etapa crónica o catatónica, el paciente necesita de medicamentos antidepresivos, e inclusive se puede llegar hasta el grado de internarlo por su seguridad, debido a que algunos presentan pensamientos suicidas que, las personas a su alrededor no pueden detectar (Organización Mundial de la Salud, 2019).

Mediante una herramienta basada en redes neuronales capaz de registrar factores que en la actualidad no se cree estén asociados al Trastorno Depresivo Mayor, y que pueda asociar y desasociar dichos factores para un diagnóstico más preciso. También se busca que, a partir de los datos de entrenamiento, pueda detectar si el paciente presenta pensamientos suicidas, con el fin de poder canalizarlo con un psiquiatra a una clínica. Estos factores pueden ser de diferente índole, ya sean culturales, sociales, o inclusive religiosos, y que no todos los pacientes presentan. Checar este párrafo no hay sentido ni correlación y quita lo que tache.

El sistema estará disponible mediante accesos controlados debido a que los usuarios deberán demostrar que cuentan con una cédula profesional que los faculte como psicólogos o psiquiatras.

# Métodos.

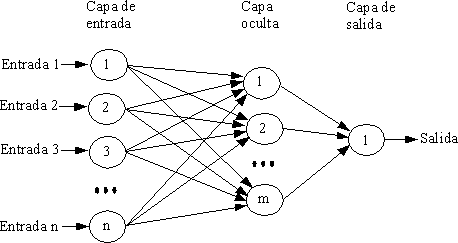
## Redes Neuronales.

Las redes neuronales son otra forma de emular ciertas características propias de los humanos, tales como la capacidad de memorizar y asociar hechos. Si se examinaran con atención todos los problemas que no son capaz de expresar su contenido en un algoritmo hay una característica en común: la experiencia. En sí, las redes neuronales no son más que un modelo artificial y simplificado del cerebro humano. (Matich, Informática Aplicada a la Ingeniería de Procesos – Orientación I, 2001, pág. 4)

Esto mismo hace que sea una herramienta útil para el sistema que queremos realizar ya que un diagnostico se basa en la experiencia previa del psicólogo o psiquiatra para dar un resultado con el cual se puede determinar si una persona tiene o no depresión y la terapia a seguir. Para el funcionamiento correcto de una red neuronal, se necesita de otra herramienta que pueda mantener los hechos almacenados, para esto mismo se crearon las denominadas bases de hechos.

Las redes neuronales pueden ser clasificadas según el tipo de aprendizaje (supervisado, no supervisado), el tipo de aplicación y la arquitectura de la conexión (monocapa, multicapa). Una de las principales características de las redes neuronales, es su capacidad de aprendizaje, a la hora de entrenar la red neuronal, se busca conseguir que una aplicación determinada, para un conjunto de entradas produzca el conjunto de salidas deseadas o mínimamente consistentes. Este proceso de entrenamiento consiste en la aplicación secuencial de diferentes conjuntos o vectores de entrada para que se ajusten los pesos de las interconexiones según un procedimiento predeterminado.

En las redes neuronales artificiales existen dos capas con conexiones, con el mundo exterior. Una capa de entrada, donde se presentan los datos a la red, y una capa de salida que mantiene la respuesta de la red a una entrada. El resto de las capas reciben el nombre de capas ocultas. La Figura 1, muestra el aspecto de una red neuronal artificial.



**Figura 1. Arquitectura de una Red Neuronal Artificial Multicapa.**

**Tipos de redes neuronales**

**Según su topología:**

1. Red neuronal monocapa o perceptrón simple:

La red neuronal monocapa se corresponde con la red neuronal más simple, está compuesta por una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos.

1. Red neuronal multicapa o perceptrón multicapa:

La red neuronal multicapa es una generalización de la red neuronal monocapa (también llamada perceptrón simple), la principal diferencia reside en que mientras la red neuronal monocapa está compuesta por una capa de neuronas de entrada y una capa de neuronas de salida, esta dispone de un conjunto de capas intermedias (capas ocultas) entre la capa de entrada y la de salida (figura 1). Dependiendo del número de conexiones que presente la red esta puede estar total o parcialmente conectada.

1. Redes neuronales de aprendizaje supervisado:

Se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un supervisor (o optimizador) que determina la respuesta que se debe generar para cada entrada. El supervisor controla la salida y si esta no es correcta, modifica los pesos de las conexiones, con el fin de que la salida obtenida se aproxime a la deseada.

1. Red Neuronal Convolucional (CNN)

La principal diferencia de la red neuronal convolucional con el perceptrón multicapa viene en que cada neurona no se une con todas y cada una de las capas siguientes, sino que solo con un subgrupo de ellas (se especializa), con esto se consigue reducir el número de neuronas necesarias y la complejidad computacional necesaria para su ejecución.

1. Red neuronal recurrente (RNN)

Las redes neuronales recurrentes no tienen una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos, con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria. Los datos introducidos en el momento t en la entrada, son transformados y van circulando por la red incluso en los instantes de tiempo siguientes t + 1, t + 2, …

Se desarrollarán 3 redes neuronales de clasificación:

1) La primera red neuronal nos permitirá clasificar si el paciente padece del Trastorno Depresivo Mayor o Trastorno Distímico, siendo la capa de salida compuesta de dos neuronas, que nos darán la probabilidad de que sea un trastorno u otro (una neurona destinada a la depresión y otra a la distimia).

2) La segunda red neuronal, nos servirá para poder clasificar los tipos de depresión que puede sufrir una persona, como vimos en el apartado de las variables, estos se dividen en 3: Atípico, Melancólico, y Catatónico.

3) La tercera red neuronal se encargará de predecir si el paciente sufre de pensamientos suicidas. Para que esta red neuronal se active, primero se deberá llegar a la conclusión de que el paciente sufre de depresión.

Las variables de entrada para las tres redes serán los datos que se recolectaron de los pacientes, como sus datos personales y los síntomas que padecen, exceptuando a aquellos que tengan que ver con ideación suicida, dado a que el enfoque de la tercera red neuronal es en esencia determinar si existe este factor.

Por medio de los pesos finales que se encuentren en las variables de entrada, podremos determinar si algún factor de tipo sociocultural, geográfico o demográfico es determinante para padecer estos trastornos, y que tanto “pesa” en el diagnóstico. Para poder realizar lo anterior, utilizaremos una sola base de hechos (dataset), la cual constará de un tensor bidimensional, desarrollada en MySQL. Esta base de hechos, permitirá el aprendizaje de las 3 redes neuronales, ya que la diferencia principal entre las 3 redes son las neuronas de salida.

## Base de Hechos.

Una base de hechos incorpora una representación de hechos referidos al dominio de aplicación del sistema. En este caso, es la espina dorsal de todo el sistema ya que aquí se guardarán las experiencias que la red neuronal deberá almacenar para poder aprender de ellas. Martínez (2006) comenta: “La base de hechos, es relevante para determinar la potencia del sistema con el que se está trabajando ya que este depende de la amplitud de los conocimientos contenidos en ella” (p. 47).

Esto nos dice que la elección de la forma en la que se pueda representar un conocimiento es fundamental para obtener un buen rendimiento del sistema. Esta relación debe permitir identificar conocimientos nuevos siendo lo flexiblemente necesario como para decidir incorporarlo a la base de hechos, eliminar un conocimiento contradictorio u obsoleto o sustituir conocimientos por otros más actualizados o precisos. (Martinez, 2006, p. 48)

## Aplicativo Web.

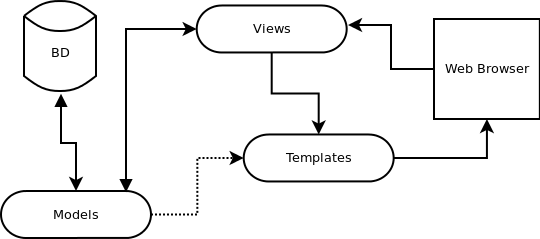
La World Wide Web (Red Informática Mundial) es un sistema de distribución de hipertexto o hipermedia interconectados y accesibles a través de Internet mediante el uso de un navegador web. Un usuario puede visualizar sitios web compuestos de archivos normalmente de extensión html que pueden contener textos, imágenes, videos u otros contenidos multimedia navegando a través de estas páginas usando hiperenlaces.

Esta manera de difusión de información ha ido en constante actualización comenzando desde las primeras páginas con formato de un boletín de periódico con el único fin de difundir noticias hasta los aplicativos más complejos como lo son las redes sociales o aplicaciones que permiten una interacción más compleja entre un usuario y el aplicativo.

Conforme el avance informático que se ha presentado mediante los diferentes lenguajes y frameworks de desarrollo web existe un mundo de posibilidades en cuanto a los procesos que se pueden automatizar para una mejor administración y seguimiento hemos concluido que esta tecnología es la indicada para el análisis de información que se desea realizar. En nuestro caso en particular, se requiere integrar Redes Neuronales Artificiales a un Sistema Web que permita el diagnostico de los trastornos previamente mencionados. Como las Redes Neuronales estarán programadas en el lenguaje Python, se necesita de una aplicación que pueda ser accedida desde cualquier dispositivo por un profesional en la salud mental para usar la Red y realizar diagnósticos un aplicativo web usando las tecnologías Mobile First daría la solución óptima y para respetar el lenguaje sobre el cual se programará la neurona se utilizará Django, framework escrito con Python.

Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente. Cuando se construye un sitio web, se necesita de un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc. Los frameworks sirven para que no se tenga que reinventar la rueda cada vez y se avance más rápido al construir un nuevo sitio.

El aplicativo web con el cual la red neuronal podrá ser alimentada y generar los reportes será programado en Python para tener una mejor conexión con la neurona que estará en el mismo lenguaje. Se utilizará el framework previamente mencionado el cual provee una estructura ideal con el patrón MVT. Para Django el Modelo-Vista-Template es similar al Modelo-Vista-Controlador utilizado en otros frameworks como Laravel o Spring Boot.



**Ilustración 1 - Modelo MVT**

El modelo es la capa de acceso a la base de datos. En esta capa contiene toda la información sobre los datos: accesos, validaciones, comportamientos y relaciones. El template (plantilla) es la capa de la presentación. En esta capa se contienen los archivos html con la estructura de cómo se mostrará el contenido del aplicativo. El view (vista) es la capa de la lógica de negocios. Dentro de esta capa se contiene la lógica que accede al modelo y la delega a la plantilla correspondiente, este es el puente entre los modelos y las plantillas.

Se tiene contemplado el uso de SQLite para almacenar la base de hechos y las bases de datos utilizadas tanto para la red neuronal como para el aplicativo web. Dentro de la base de datos para el aplicativo, las tablas principales para que el sistema funcione correctamente deben de incluir los diagnósticos generados, el historial de usuarios y los usuarios registrados con sus respectivas relaciones para la gestión de los resultados de la red.

El aplicativo manejara 3 tipos de usuarios:

1.- Anónimo: Todo usuario sin registro previo, pueden visualizar las páginas de registro, inicio de sesión y la página principal del aplicativo.

2.- Doctores: Usuarios que hayan sido registrados con éxito demostrando su profesión con la cédula profesional, pueden visualizar la respectiva página de inicio, el historial de diagnósticos realizados por mes/año, la ventana para realizar un nuevo diagnóstico, la página de modificación de credenciales y un apartado estadístico de la cantidad de pacientes que ha diagnosticado y se haya encontrado uno de los trastornos.

3.- Administradores: Los creadores del aplicativo o personal encargado de la administración de la base de datos. Tendrá acceso a una página donde se muestre una tabla con los usuarios registrados en el sistema por orden alfabético donde podrá editar, eliminar o restringir privilegios a los usuarios para mantener una gestión de las cuentas activas.

Se utilizará Ajax para realizar peticiones (get, post, put, delete), Bootstrap para los estilos y JQuery para la interacción que tendrá el usuario con el aplicativo. Mediante estas librerías se planea utilizar una interfaz llamativa e intuitiva para el usuario buscando que todo quede como una Single Page Application (SPA).

# Desarrollo.

## Aplicativo Web.

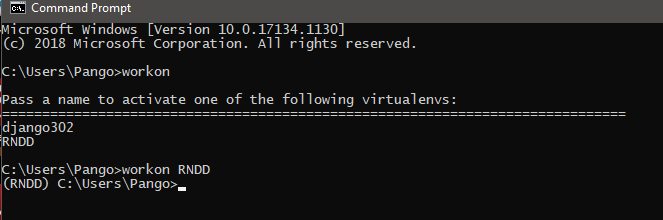
**Creación del Proyecto**

Para poder utilizar Django se necesita tener instalado Python y PIP en el sistema operativo donde se valla a desarrollar, para esto basta con entrar a la página oficial de Python ([www.python.org](http://www.python.org)) ir a la sección de descargas y posteriormente descargar e instalar la última versión de Python ejecutando el archivo ejecutable obtenido de la página oficial de Python.

El otro componente necesario para las librerías y dependencias que se vayan a ocupar en el proyecto es PIP. Para poder instalar PIP basta con ejecutar el comando *python get-pip.py* dentro de la consola de comandos en un sistema operativo que ya tenga instalado Python previamente.

Python tiene un manejador de entornos virtuales, pueden existir N número de estos los cuales pueden configurarse de manera especifica para cada proyecto creado, estos servirán para levantar el servidor del proyecto para manejar una configuración adecua dependiendo de la versión del framework o del lenguaje en la que se este desarrollando. Para esto se tiene que ejecutar el comando *pip install virutalenvwrapper-win* en la consola y esto se encargara de descargar y instalar las paqueterías necesarias. Posteriormente se necesitara instalar workon librería la cual permitirá usar los diferentes entornos virtuales creados con la paquetería anterior.

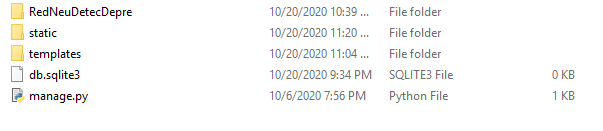
Para crear un entorno se utiliza el comando *mkvirtualenv “Nombre del proyecto sin comillas”* . Se puede ejecutar el comando *workon* para listar los entornos actuales y se debe entrar al entorno actual escribiendo *workon “nombre del proyecto sin comillas”.*



**Ilustración 2 - Comandos para la creación de entornos virtuales**

Una vez se este dentro del entorno virtual, todas las librerías y paqueterías que se vayan a utilizar y deban ser instaladas al proyecto, se instalaran en el entorno virtual encargado del levantamiento del servidor local del mismo. Para poder usar Django se debe de instalar, esto se hace ejecutando el comando *pip install django* dentro del entorno virtual. Posteriormente se debe de crear un proyecto en django ejecutando el comando *django-admin startproject “nombre del proyecto sin comillas”*. Esto será lo necesario para una configuración básica de un proyecto en blanco en django.

Se habrá creado una nueva carpeta con los archivos y carpetas necesarias para el framework como en la Ilustración 3. La carpeta *static* contendrá los archivos de estilo .css y los archivos de funcionalidades .js al igual que las librerías que se vayan a incorporar. *Templates* tendrá los archivos .html que tendrán la estructura de como se irán mostrando los datos para el usuario. *RedNeuDetecDepre* en este caso es el nombre del proyecto y esta carpeta tiene una estructura interna que se explicara posteriormente. El archivo *db.sqlite3* es el encargado de la base de datos, aquí se crearan las tablas para los usuarios, diagnósticos y resultados junto con sus respectivas conexiones. Por último, *magane.py* es el archivo que utiliza django para compilar y ejecutar el servidor local con el proyecto, este mismo contiene comandos los cuales configuran ciertas opciones del proyecto.



**Ilustración 3 - Estructura básica de un proyecto en django**

Dentro de la carpeta con el nombre del proyecto se encuentran los archivos de la ilustración 4. Los archivos que se usarán principalmente son *views.py*, *urls.py*, *settings.py*.

*Views.py*

Este archivo es el que contiene toda la lógica que se manejará en la página. Una ventaja de Django es que permite separara los módulos mediante *apps*. Estás serán explicadas más adelante, por lo pronto en este archivo se llevara la lógica para el módulo principal del aplicativo el cual tendrá las funciones de inicio de sesión y registro. Tomando como referencia a un framework que maneje MVC, este archivo sería el controlador, pero en este caso es llamado la *vista*.

*Urls.py*

Para que la lógica se ejecute primero tiene que llamarse un link/hipervínculo/url, estás serán declaradas dentro de este archivo. Dichas urls contienen el hipervínculo que tendrá que seguir el navegador para acceder, la vista a la que hará referencia y el nombre que se le dará al link.

*Settings.py*

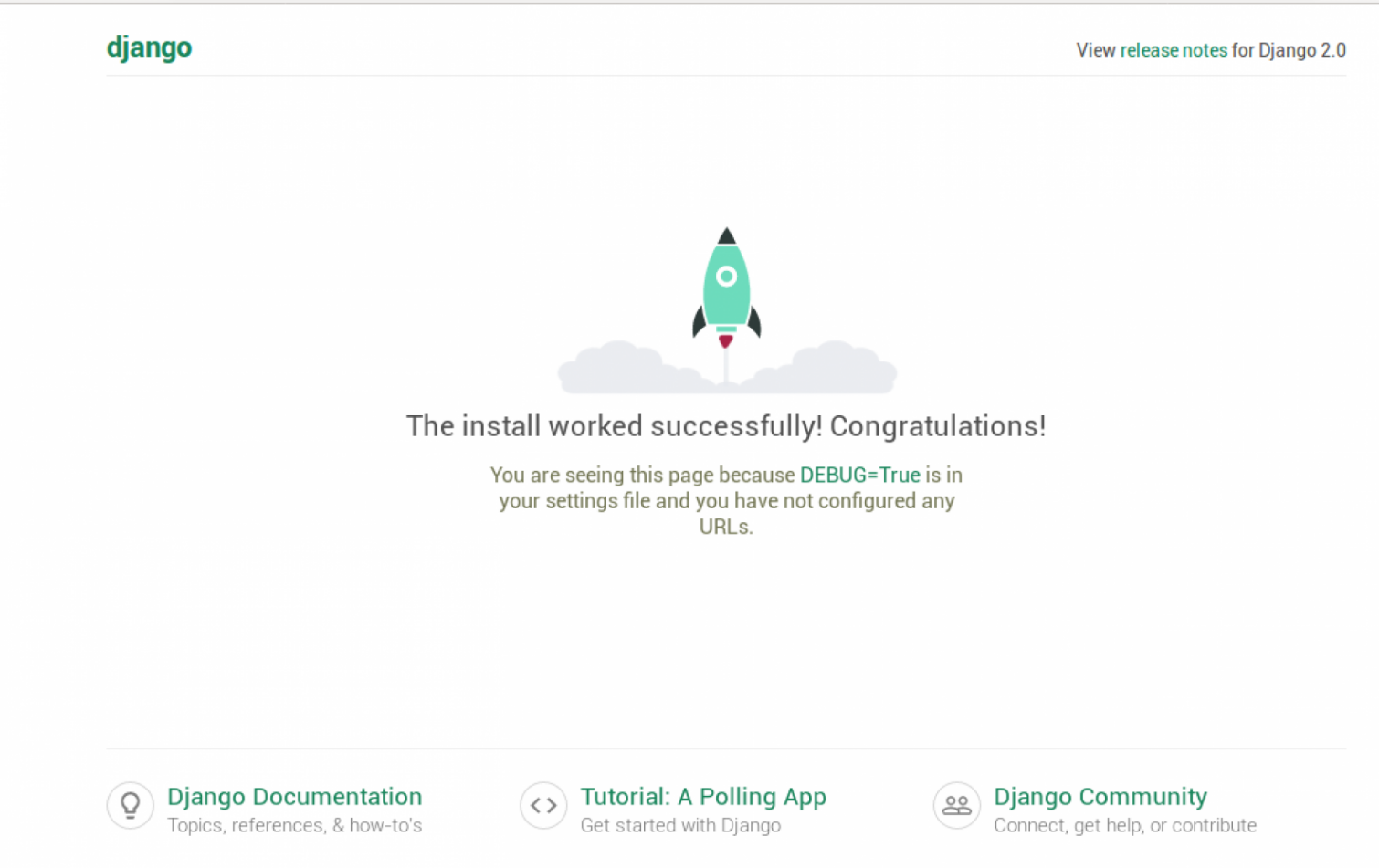
Por ultimo esta el archivo encargado de las configuraciones globales para el proyecto. Aquí se define si el proyecto sigue en modo de desarrollo o producción, se importan las librerías descargadas para usar y las rutas para la carpeta static y templates que normalmente no vienen programadas.

Este conjunto de archivos serán los encargados para el modulo que se encargara de los registros e inicio de sesión para los usuarios, al igual que la protección de los links a los cuales solo tendrán accesos las personas que ya tengan credenciales válidas para el aplicativo.

Dicho conjunto se puede repetir dependiendo de los módulos que se vayan a manejar, de esta manera se mantiene la lógica separada para cada sección que se vaya a necesitar. En el caso del aplicativo que se va a desarrollar, necesitamos de dos módulos más. Uno se encargará de los componentes necesarios para los doctores y otro se encargará de lo necesario para los administradores.

Para poder crear los módulos previamente mencionados, se tienen que ejecutar en la consola de comandos que tiene el entorno las siguientes instrucciones: *python manage.py startapp doctores* y *python manage.py startapp administradores* y luego registrarlas dentro del archivo *settings.py*. Se crearan las carpetas respectivas con los mismos archivos que el modulo de inicio. De esta manera se manejaran urls diferentes y vistas diferentes para cada módulo, facilitando la programación del aplicativo.

Una vez se tengan las configuraciones anteriores, se puede probar compilando el proyecto verificando que todas las configuraciones se hayan cargado de manera correcta y no exista algún error en la sintaxis. Esto se hace ejecutando el comando *python manage.py runserver* (todos los comandos deben ser ejecutados en la carpeta raíz del proyecto, justo donde se encuentra el archivo *manage.py*). Al ser un framework interpretado no se necesita compilar cada cambio, cada que se guarda un documento la consola se reinicia y revisa que todo este correcto, de lo contrario muestra donde se encuentra el error y no permite cargar el aplicativo. De todo estar correcto se mostrará un pantalla de inicio como en la ilustración 4.



**Ilustración 4 – Página de inicio predetermina Django**

Lo siguiente es crear los *objetos* que se encargaran de almacenar los datos de los usuarios, los diagnósticos y los resultados de estos. Hay un concepto en el mundo de la programación llamado *programación orientada a objetos*. La idea es escribir las tablas como un conjunto de propiedades y acciones (tales como se haría en, por ejemplo, java). Estas tablas que ahora se llamarán *objetos* estarán dentro del archivo *models.py* dentro del modulo respectivo. El modulo de usuarios tendrá los *objetos* para las tablas de registro y roles para los usuarios, doctores tendrá los diagnósticos y los resultados. Estos archivos tendrán la siguiente sintaxis:

from django.db import models

from django.utils import timezone

class ”Nombre del modelo sin comillas” (models.Model):

dato1 = models.ForeignKey('tabla/objeto de donde se obtiene la llave', on\_delete=models.CASCADE) //esto es en caso de que exista una relación con algúna tabla que requiera de una llave foranea

dato2 = models.CharField(max\_length=200)

dato3 = models.TextField()

created\_date = models.DateTimeField(

default=timezone.now)

published\_date = models.DateTimeField(

blank=True, null=True)

def “Nombre de función sin comillas”(self):

self.published\_date = timezone.now()

self.save() //en este caso devuelve la fecha actual para el campo published\_date

Con todos los modelos registrados en sus respectivos archivos hay que agregar estos modelos a la base datos. El primer paso es hacer saber a Django que se han realizado o agregado los nuevos modelos. Esto se hace entrando a la terminal y escribiendo el comando *python manage.py makemigrations “nombre de la app sin comillas”*, en este caso, el comando se ejecutara dos veces, uno para cargar los modelos del módulo de inicio (*python manage.py makemigrations RedNeuDetecDepre)* y el otro para el módulo de doctores (*python manage.py makemigrations doctores)*.

Con este comando django ha preparado archivos de migración los cuales tienen que ser aplicados a la base de datos mediante el comando *python manage.py migrate “nombre de la app”* (en este caso se sustituiría el nombre por *RedNeuDetecDepre* y luego por *doctores*). De esta manera ya se tendrían los modelos dentro de la base de datos.

Django tiene un administrador propio para las bases de datos, este trae su propio apartado visual para la gestión de las tablas de esta. Primero se debe de crear el archivo *admin.py* en cada app creada donde se registraran las tablas migradas. La sintaxis es de esta manera:

from django.contrib import admin

from .models import “NombreDelModelo”

admin.site.register(“NombreDelModelo”)

Para ver el modelo en la sección gráfica se debe de crear un super usuario, esto se realiza ejecutando el comando *python manage.py createsuperuser* y posteriormente llenar los campos solicitados escribiendo la respuesta a lo que solicite el *wizard* dando “enter” tras cada entrada. Los campos que solicita son los siguientes:

Username: ola

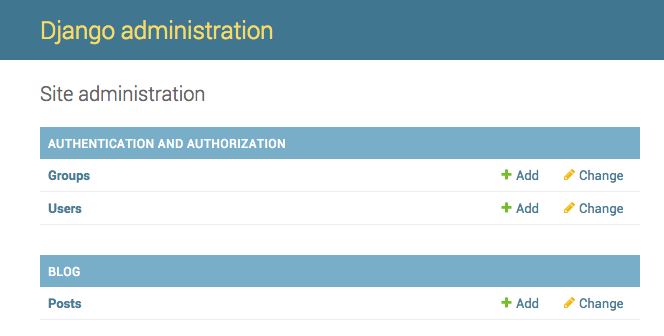
Email address: ola@example.com

Password:

Password (again):

Superuser created successfully.

Una vez se tenga esto se debe de ingresar mediante la url que otorge el comando *python manage.py runserver* de la siguiente manera: *url/admin*. Dara un formulario para el inicio de sesión el cual tras ser llenado mostrara una interfaz con las tablas creadas tras la migración. Aquí es donde se podrá ver de manera directa los datos que se van llenado en las tablas, básicamente esta será la pantalla del administrador adaptando los estilos a los colores y logos seleccionados para este proyecto.



diganosticos

DOCTORES

Aquí el administrador podrá directamente hacer modificaciones a los datos en la base, tanto agregar a nivel sql como eliminar entre otras operaciones.

Estas serán las configuraciones para la estructura lógica y funcional del proyecto. Las pantallas prototipo serán creadas utilizando la herramienta JustInMind Prototyper para luego ser pasadas a la estructura html, agregar estilos con los archivos css y agregar las funcionalidades en las vistas de Django.

# Conclusiones.

# Créditos.

# Referencias Bibliográficas.

# Información de los autores.