**YUKIHIRA BOT (API)**

**-------------------------------Commands.py------------------------------------**

Variables generales:

A continuación se enumerarán las variables que la mayoría de las funciones de este módulo reciben como argumentos.

*-state:* un **integer** que puede ser 0 o un número de dos cifras del 10 hasta 80, el primer digito indica el comando que se está ejecutando, mientras el segundo indica la altura en la que se encuentra el usuario dentro del proceso de dicho comando.

*-command:* un **string** que representa el comando que el usuario quiere ejecutar. Un comando se distingue debido a que el **string** comienzacon un *slash* (/).

*-id:* un **integer de 9 dígitos** con el valor del id del usuario provisto por la API de Telegram.

*-username:* un **string** con el valor del nombre del usuario provisto por la API de Telegram. Se denota siempre al principio con un arroba(@) por convenciones de la plataforma Telegram.

*-firstname:* un **string** con el valor del nombre de pila del usuario provisto por la API de Telegram.

*-type:* un **string** que identifica el tipo de chat por el cual un mensaje fue recibido, puede ser *private* o *group.*

*-text:* un **string** del texto plano en sí del mensaje enviado por el usuario.

Funciones:

* identifyCommandByState(*state*):

Identifica un comando dado el *state* provisto. Redirige al comando identificado por el primer digito del *state*, y la altura será determinada dentro de la función del comando al que se redirigió.

* identifyCommand(*command, state, id, username, firstname, type, text*):

Identifica un commando a ejecutar dado un argumento *command* de tipo string.

Redirige a la llamada de la funcion asociada al valor del comando, y un caso de excepción en el cual el comando recibido no exista.

* start(*id, username, firstname*):

Empieza la interaccion con el bot, se almacenan el id del usuario, su nombre y su username en la base de datos.

* help(*id*):

Muestra la lista de comandos que realiza el bot.

* NewRestaurant(*state, id, username, firstname, type, text*):

Registra los estados (en este caso el nombre y tipo del restaurant) en la base de datos para saber en qué comando y altura esta

* RestDescription(*state, id, username, firstname, type*):

Registra una descripción de máximo 400 caracteres del restaurant

* EditMenu(*state, id, username, firstname, type*):

Permite agregar o eliminar platillos al menu del restaurant a seleccionar

* EditRecipe(*state, id, username, firstname, type*):

Agrega los ingredientes a utilizar en el platillo seleccionado

* DishDescription(*state, id, username, firstname,* *type*):

Agrega una breve descripcion del platillo

* NewOrder(*state, id, username, firstname, type*):

Se genera una orden, el usuario escoge la categoria o tipo de comida que desea, el restaurante deseado, se mostrara el menú que ese restaurant proporciona, y el usuario escogerá los platillos que desee

* cancel(*id*):

Detiene cualquier funcion que este siendo ejecutada en el momento

----------------------------Dbconnection.py-----------------------------------

* connect():

Se intenta la conexión a la Base de datos.

* **getUserState**(*id*):

Se extrae el id del usuario de la base de datos

* saveUserState(*id, state*):

Se implementa el id y el estado del usuario a la base de datos

---------------------------messageHandling.py---------------------------------

* botHolder(*bot*):

La variable bot se vuelve global, haciendo que su alcance sea a todo el programa

* handle(*msg*):

Recolecta y analiza la información del usuario cuando este envía un mensaje, además del tipo chat y la hora en la que se envió dicho mensaje

* Analize(*msg*, *text, id, type*):

Se analizan los datos que se están capturando

* sendMessage(*id*, *text*):

El bot envía un mensaje de texto al usuario

**MINUTAS**

**Registro del 29/09/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio.

Tema(s) tratado(s):

Discusión final sobre proyecto a desarrollar. Bot de telegram que asista una plataforma de recomendaciones de platos posibles para renovaciones de menú, para un restaurant de comida rápida. Posible módulo de pedidos

a domicilio personalizados.

Para próxima semana:

Diagrama entidad relación o diagrama de la BD. Finiquitar detalles de plataforma a utilizar.

**Registro del 06/10/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio.

Tema(s) tratado(s):

Entrega de la base de datos. Mejoras acerca de las unidades y la cantidad de ingredientes en la tabla de receta. Formas en las que el bot se comunicaría con el usuario (teclado especial y comandos).Decisión final de plataforma a usar.

Para próxima semana:

Aplicar cambios en base de datos y comprobar que funcione.

**Registro de la Reunión 20/10/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas (Desarrollador, Arquitecto), Victor Panunzio(Arquitecto, Documentista), Carlos Hernández(Desarrollador, Documentista).

Tema(s) Tratado(s):

Almacenamiento de las recomendaciones de recetas en una misma tabla. Creacion de nueva tabla de relación dishes y users para manejar distintos restaurantes. Hacer un campo array o JSON para las categorías favoritas del usuario.

Para próxima semana entregar:

1. Diagrama Entidad Relación final. (Luigi)

2. Nomenclatura de los comandos del bot. (diagrama de interacción) (Victor)

3. Diagrama de casos de uso UML. (Carlos)

**Registro de la Reunion 3/11/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio, Carlos Hernandez.

Tema Tratado:

Estructura de la función de Recomendaciones, el programa recomendara a base de la categoría del platillo, al ver la lista de órdenes, la recomendación vendrá de las categorías más escogidas por los usuarios, y de dicha categoría recomendar el platillo más pedido.

**Registro de Reunión 10/11/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio, Carlos Hernandez.

Tema(s) Tratado(s):

Se discutió el tema de los módulos pendientes a realizar, se entregó otra versión del diagrama de caso de uso, se define para la semana entrante un adelanto en la conexión de la base de datos y mayor información sobre los comandos del bot.

**Registro de Reunión 17/11/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio, Carlos Hernandez.

Tema(s) Tratado(s):

Se entregaron varios de los comandos terminados en la semana anterior, se estableció que la recomendación del platillo será un comando parte que solo se realizara cuando el usuario la pida.

Para la próxima semana:

Se entregaran el resto de los comandos y se trabajara en el algoritmo para las recomendaciones.

**Registro de Reunión 24/11/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio, Carlos Hernandez.

Tema(s) Tratado(s):

Se estableció la base del algoritmo para las recomendaciones con la fórmula de media poblacional.

Para la próxima semana:

Se entregaran el resto de los comandos hechos y se establecerán los parámetros para la documentación final del proyecto.

**Registro de Reunión 30/11/16**

Presentes: Ing. Rainier Araujo, Luigi Rojas, Victor Panunzio, Carlos Hernandez.

Tema(s) Tratado(s):

La documentación final, que corresponderá a la segunda y última entrega del proyecto, para la próxima semana, estará compuesta por:

API, comprendida por los métodos implementados en el bot.

Diagramas de: Entidad-Relación, Caso-Uso, Despliegue.

Minutas.

**Diagrama de Despliegue**

V

Ordenador

Servidor de App

Cliente de Telegram (Escritorio o Navegador)

HTTP/HTTPS

Telegram

HTTP/HTTPS

Internet

API de Telegram

Dispositivo Movil

HTTP/HTTPS

App

Servidor de Base de Datos

PostgreSQL

Diagrama de Caso de Uso

