

Pokedex Discord BOT



Discord



1.Introducción.....	4
1.1.Motivos de elección.....	4
1.2. Descripción.....	4
1.3.Pokedex.....	4
1.4. Discord.....	5
1.5. Bot Discord.....	5
1.5.1 Ejemplos de BOTs de Discord.....	5
MEE6.....	5
Coin Flipper.....	5
Dank Memer.....	6
Tatsu.....	6
Friend Time.....	6
HangMan.....	6
ServerStats.....	6
1.6.Objetivos.....	6
1.7. Aplicaciones prácticas.....	6
1.6.Estudio de mercado.....	7
Pokemon.....	7
Pokebest.....	8
Pokedex bot.....	9
2. Fundamentos generales teóricos y prácticos.....	10
2.1. Parte de base de datos.....	10
2.2. Parte del BOT.....	10
3.Creación del BOT.....	11
4.Creacion del proyecto python.....	13
4.1. Primeros pasos.....	13
4.2. Crear un evento.....	14
4.3. Crear un comando.....	14
Sin parametros.....	15
Con parametros.....	15
5.Creación de la base de datos.....	15
5.1. Tablas.....	15
5.1.1. Pokemon.....	15
5.1.2. Types.....	16
5.1.3. Poke_Type.....	16
5.1.4. Region.....	16
5.1.5. Users.....	16
5.1.6. Caught.....	16
5.1.7. Type_Effectiveness.....	17
5.2. Introducción de los datos básicos.....	17
6. Funciones bot.....	17
6.1. Buscar por número de pokédex.....	17
6.2. Buscar por nombre.....	18

6.3. Buscar según la región.....	18
6.4. Registrarse.....	18
6.5. Capturar pokemon.....	18
6.6. Ver progreso total.....	18
6.7. Ver progreso en una región.....	19
6.8. Ver pokemons restantes en total.....	19
6.9. Ver pokemons restantes en una región.....	19
6.10. Consultar efectividad.....	19
6.11. TypeChart.....	20
6.12. Whos that pokemon.....	20
6.13. Info.....	20
7. Recomendaciones para el año que viene.....	20
7.1. Acceso a datos.....	21
7.2. Interfaces.....	21
7.3. Extra.....	21
8. Conclusiones.....	21
9. Bibliografía.....	22
10. Recursos utilizados.....	22

1.Introducción

1.1.Motivos de elección

Discord es la red social que más utilizo actualmente debido a lo sencillo que es usarla, y a la gran variedad de funcionalidades extra que nos ofrece sobre su competencia. Una de las más destacables es poder crear y unirse a servidores

Los servidores permiten a incontables usuarios interactuar entre sí en el mismo lugar. Pero lo más destacable de estos servidores son los bots. Siempre me ha llamado la atención la gran variedad de funciones completamente distintas que estos pueden tener. Es por esta curiosidad que he decidido crear un bot propio para esta plataforma

A la hora de añadir un bot a un servidor. Lo más fácil es buscar BOTs creados por los desarrolladores de discord para añadirlos a tus servidores. Pero eso no siempre sirve puesto que a veces ese bot no cuenta con las funcionalidades deseadas. Es por esto que se nos permite crear tu BOT desde cero para luego añadirlo a tu servidor. Y por supuesto si lo haces público, cualquier persona podrá añadir tu BOT a su servidor

1.2. Descripción

Este bot de discord tendrá la misma funcionalidad que una pokédex original, incluso con algunos extras, se utilizará la librería de terceros discord.py, la cual conecta directamente con el API de discord, permitiendo así que podamos crear nuestro bot.

Hay varios conceptos que es altamente posible que deban ser explicados, como pokedex y discord. Los cuales explicaré a continuación

1.3.Pokedex

La Pokédex es una enciclopedia virtual utilizada en el mundo ficticio de Pokémon. Es una herramienta importante para los entrenadores Pokémon, ya que les proporciona información detallada sobre todas las especies de Pokémon. La Pokédex muestra datos como la descripción, el tipo, las estadísticas de combate etc

En los juegos de Pokémon, los entrenadores reciben una Pokédex vacía al comenzar su aventura y su objetivo es completarla capturando todos los Pokémon que existen

La Pokédex es de gran utilidad ya que muestra a los entrenadores toda la información que puedan desear de un Pokémon específico

1.4. Discord

Discord es una red social pensada principalmente para la comunidad Gamer, ya que todas sus funcionalidades permiten jugar, transmitir y comunicarse entre muchas más cosas a distancia de forma muy sencilla. Aunque estas funcionalidades también son útiles para cualquier grupo de personas, esto engloba incluso empresas o colegios

Discord nos permite crear servidores. Estos servidores pueden ser tanto públicos como privados y pueden tener diferentes configuraciones, como roles para organizar los permisos de todos los miembros del servidor. Un servidor te permite comunicarte a tiempo real con un gran número de personas al mismo tiempo, lo puedes hacer por escrito o por un chat de voz

Además también puedes añadir funcionalidades extra a tus servidores de varias formas distintas como por ejemplo los bots. Estos son pequeños programas que mejorarán nuestra experiencia mediante nuevas y mejoradas funciones.

1.5. Bot Discord

Un bot de Discord es un programa diseñado para mejorar la experiencia de los usuarios en los servidores. Los bots se ejecutan en servidores y se conectan a través de la API de Discord, lo que les permite realizar diversas funciones y automatizar tareas.

Los bots pueden tener muchas funcionalidades y características, dependiendo de su propósito. Algunos bots están diseñados para automatizar la administración de servidores, realizando tareas como la gestión de roles, la prohibición de usuarios, la eliminación de mensajes inapropiados etc

Otros bots se enfocan en proporcionar entretenimiento y diversión a los usuarios. Pueden tener comandos para reproducir música, mostrar memes, juegos o incluso tener funciones de economía virtual.

En resumen, los bots de Discord son programas o aplicaciones que agregan funcionalidad adicional a los servidores de Discord

1.5.1 Ejemplos de BOTs de Discord

MEE6

[The Best Discord Bot for Your Server | MEE6](#)

Coin Flipper

[Coin Flipper - Discord Bots](#)

Dank Memer

[Dank Memer | Home](#)

Tatsu

[Tatsu - Build a Fun Discord Community](#)

Friend Time

[Friend Time - Discord Bots](#)

HangMan

[Hangman - Discord Bots](#)

ServerStats

[ServerStats Bot - Home](#)

1.6. Objetivos

El objetivo principal de este proyecto es crear una aplicación que ofrezca las mismas funcionalidades que las Pokédex originales de los videojuegos

En las entregas más recientes se ha visto recortada de manera astronómica la cantidad de Pokémons registrables en la Pokédex, dificultando así la aventura de los que quieren atrapar a todos los Pokémon, y no solo a los que son registrables en esa entrega

Es por eso, que esta Pokédex no se quedará en mostrar los pokémon que habitan en la región de la última entrega, sino que mostrará la información de un total de 809 especies de Pokémons distintos, lo que viene a ser todos menos los de la entrega más reciente.

Ademas tambien podras obtener informacion acerca de las regiones, los tipos y cuan efectivo es hacia otro pokémon, e incluso que puedas seleccionar 6 pokemons para ver si seria un equipo factible

1.7. Aplicaciones prácticas

Desde las últimas entregas de la franquicia de pokémon, se nos impide registrar a todas las criaturas al 100% ya que decidieron reducir la pokédex para cada entrega de ahora en adelante.

Hay una alternativa llamada Pokémon Home, pero para poder usarla de manera eficiente, es necesario pagar una mensualidad por ello, motivo por el cual no es la opción preferida de

muchos Debido a esta división de las criaturas, la tarea de llevar una cuenta o registro de cuantos pokemons tienes o cuales te faltan por capturar se vuelve muy tediosa

Es aquí donde entra el uso de mi bot, el cual les dejará llevar un registro de absolutamente todos los pokemons.

También es posible que necesite saber las debilidades de un pokemon es específico, situación que también se resolvería con el uso de mi bot

Y el uso más evidente, es poder ver la información acerca de un pokémon específico, o varios a la vez

1.6. Estudio de mercado

He decidido hacer este tipo de bot debido a que discord es actualmente la plataforma de mensajería online más usada por la comunidad gamer. Esto permitirá que si en un futuro quisiera publicar mi proyecto, podría llegar a ser muy bien recibido ya que está desarrollado para la plataforma que definitivamente usan la gran mayoría de los fans de pokemon

He hecho una investigación sobre los distintos bots similares que existen actualmente en discord, para ver qué funcionalidades pueden hacer que los usuarios elijan mi bot antes que el de los demás.

Pokemon

Este bot trae funcionalidades parecidas a las de un juego, puedes empezar partida (lo que en el mio seria la funcionalidad de registrarse). Una vez iniciada partida puedes hacer muchas cosas que en un juego original, como seleccionar un pokémon inicial, captura, evolucionar e intercambiar pokemons, los cuales aparecen aleatoriamente, también es posible comprar objetos

Tiene funcionalidades muy distintas al mio debido a que este bot está más destinado a crear una experiencia de juego que a brindar información al usuario

Tra haber probado las funcionalidades similares al mío he encontrado las siguientes diferencias:

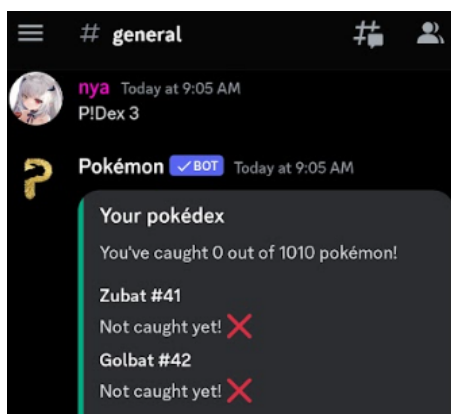
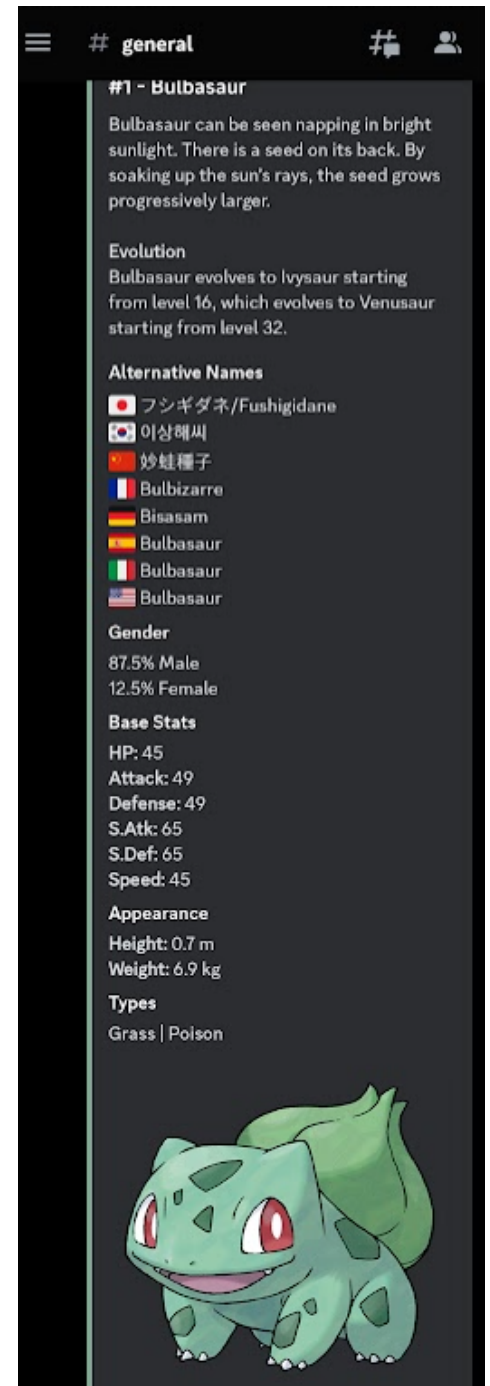
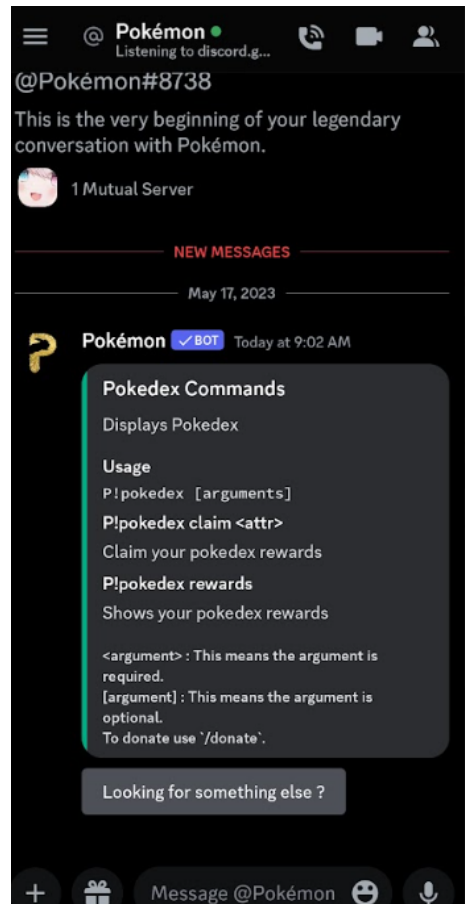
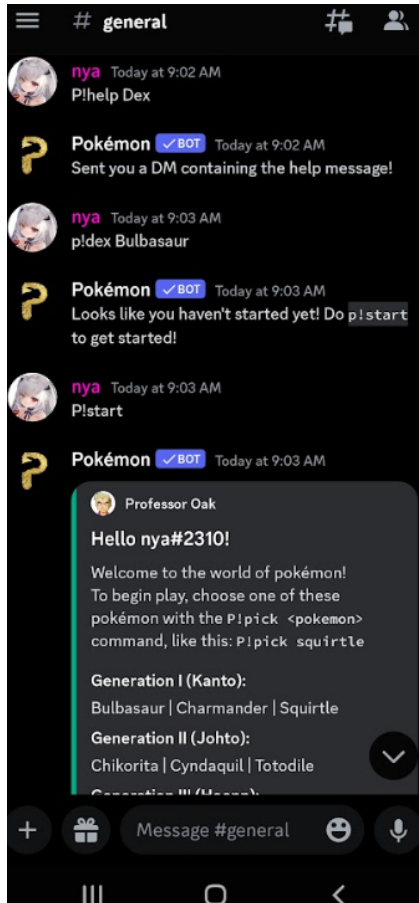
- El bot investigado es poco intuitivo. Dispone de un comando de ayuda el cual nos hace pensar que solo hay un único comando para ver la información de los pokémon, mientras que tiene dos

- Es inconveniente por que para realizar cualquier acción por sencilla que sea, necesitas estar registrado

- Muestra algo más de información, aunque el 90% de ella no es útil

-De entre la información que muestra, lo que me gustaría añadir a la mía en el caso de ser factible en el tiempo, sería la información de evolución

En las siguientes capturas de pantalla se puede apreciar lo dicho en los puntos anteriores



Pokebest

En la información que proporcionan en su página web muestra que tiene funcionalidades parecidas al anterior, solo que este no tiene nada acerca de la pokédex. Por ese motivo si alguien quiere un bot de combates, se decantaría por este, pero si lo que busca es obtener información, o poder llevar un registro del progreso de su pokédex, se decantaría por el mio

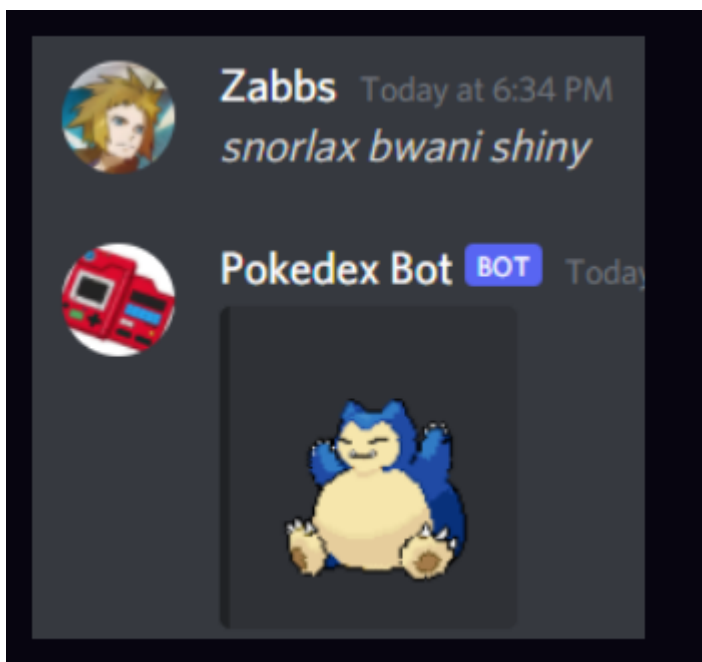
Pokedex bot

Tiene un nombre similar al mio, aunque las funcionalidades cambian. En su página oficial nos muestra sus comandos, y capturas de pantalla acerca del resultado de algunos de ellos. Sus comandos son los siguientes:

This bot is a dream come true for competitive Pokemon players. It has multiple Pokedex features
It can give information on the following pokemon stuff-

- Items (`dexy item <item>`)
- Abilities (`dexy ability <ability>`)
- Pokemon (`dexy dex <pokemon>`)
- Lightweighted Pokemon information (`dexy ldex <pokemon>`)
- Moves (`dexy move <move>`)
- Sprites (Italics and underscores work too as shown)
- Artwork (`dexy artwork <pokemon>`)
- Weaknesses(`dexy weak <type1> [type2]`)
- Movesets (`dexy moveset <pokemon> <version name or initials> [learn method]`)
- Evolution Chains (`dexy evo <pokemon>`)
- Aliases (`dexy alias add "<alias name>" <pokemon>`)

Por ejemplo, cuando buscas un pokemon, solo te mostrara el sprite y no información acerca del mismo



Debido a que esto es una investigación, he probado el bot para ver si de verdad solo muestra el sprite cuando buscas el pokemon. Aunque si usas el comando `dexy dex <Nombre>` sí que muestra información de forma parecida al primer bot investigado, no es

posible buscar por número, funcionalidad que es bastante útil y debería ser indispensable en una pokédex

En cuanto a las ventajas que tendría sobre el mío, se verían casi anuladas debido a que no muestra información acerca de ellas en su página oficial

2. Fundamentos generales teóricos y prácticos

Como ya he explicado antes la Pokédex es una enciclopedia, lo cual es el equivalente a una base de datos, por lo que mi proyecto se dividirá en dos partes

1. Conexión y operaciones en una base de datos. En esta parte se almacenará en la base de datos toda la información que vaya a necesitar para mi proyecto. Y también recogeré dicha información para dejarla lista para ser usada
2. Usar la base de datos desde cualquier servidor, en cualquier parte del mundo mediante el BOT. En esta parte es en la que le daré un uso a toda esa información

2.1. Parte de base de datos

Esta parte es la más sencilla ya que la mayor parte de lo que usaré se ha estudiado en el módulo de Acceso a Datos.

Para este proyecto en lugar de usar una base de datos temporal H2 Console, crearemos una conexión a una base de datos de phpmyadmin.

El lenguaje de programación utilizado será Java

2.2. Parte del BOT

Esta es la parte más complicada ya que todo lo que se necesita es completamente nuevo y hay que investigar y probar todo de cero hasta conseguir crear las funcionalidades deseadas para mi Pokédex

El lenguaje de programación a utilizar será Python, debido a que una de las librerías más completas para programar bots de discord es Python.

También es común crearlos usando JavaScript, pero usaremos Python debido a que JavaScript es parte del ciclo de Desarrollo de Aplicaciones Web (DAW) y no del ciclo de Desarrollo de Aplicaciones Multiplataforma (DAM),

Cabe destacar que el bot no se crea desde el código, desde el código creamos la funcionalidad. Es por eso que primero tendremos que crear el bot en sí, un bot recién creado es esencialmente un contenedor.

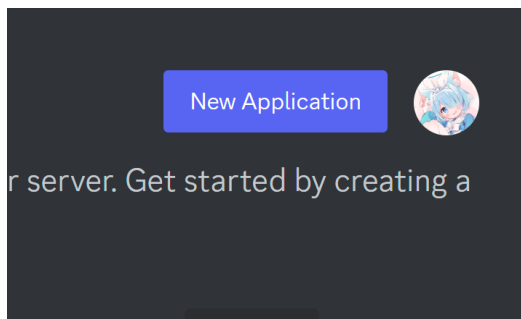
3.Creación del BOT

Para crear un bot, seguiremos los siguientes pasos:

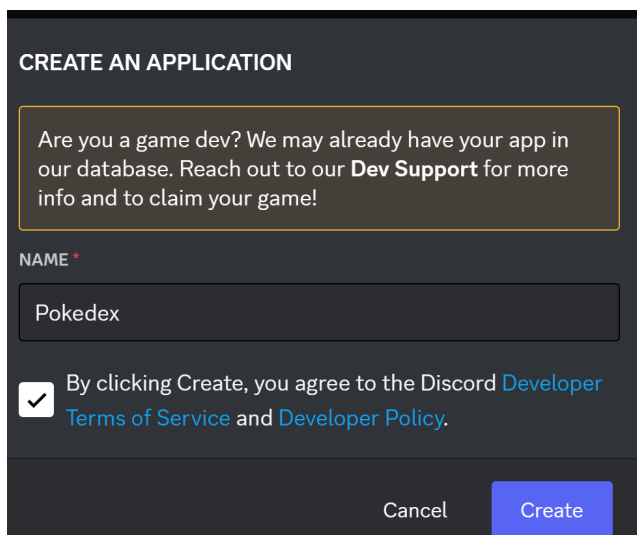
Lo primero es acceder a la página oficial dedicada para ello:

- <https://discord.com/developers/applications>

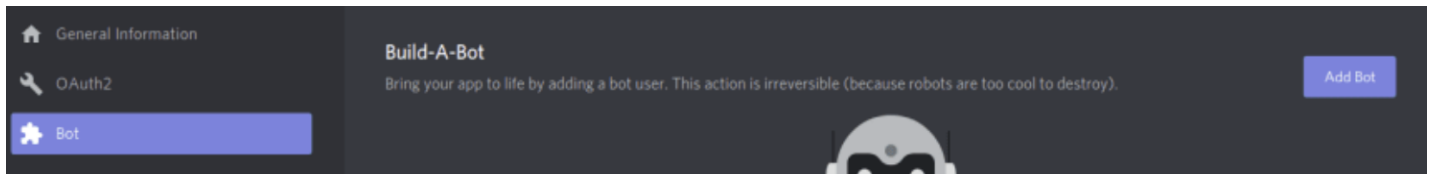
Una vez hayamos accedido con nuestro usuario, pulsaremos en el botón “Crear Aplicacion” situado en la esquina superior izquierda



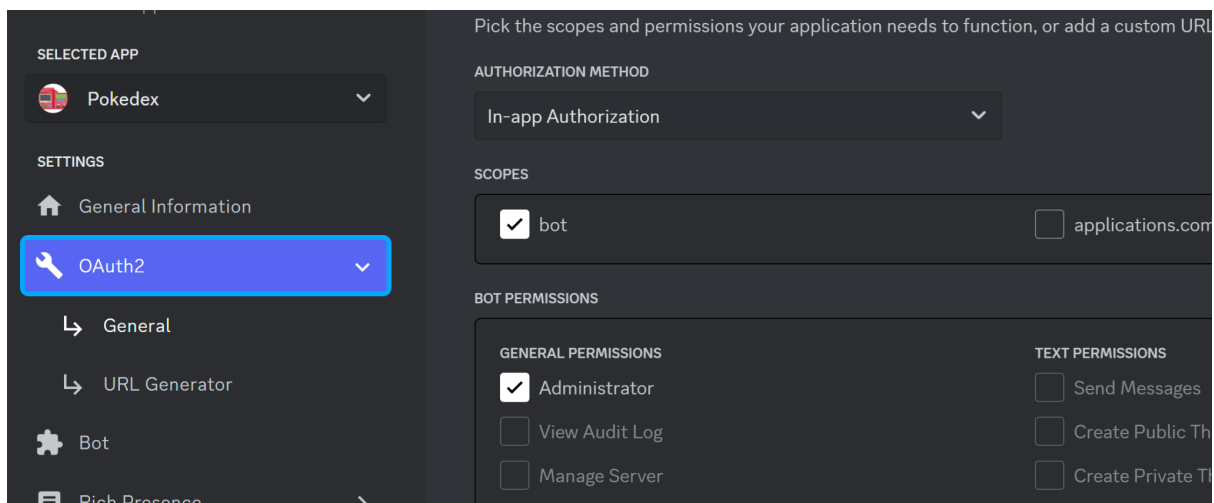
Le damos un nombre, aceptando los términos de uso

A screenshot of the 'CREATE AN APPLICATION' form in the Discord developer portal. The form has a dark background. At the top, it says 'CREATE AN APPLICATION'. Below that, there is a yellow box with text: 'Are you a game dev? We may already have your app in our database. Reach out to our **Dev Support** for more info and to claim your game!'. Underneath, there is a label 'NAME *' followed by a text input field containing 'Pokedex'. Below the input field, there is a checkbox with a checkmark and text: 'By clicking Create, you agree to the Discord [Developer Terms of Service](#) and [Developer Policy](#).' At the bottom right, there are two buttons: 'Cancel' and 'Create'.

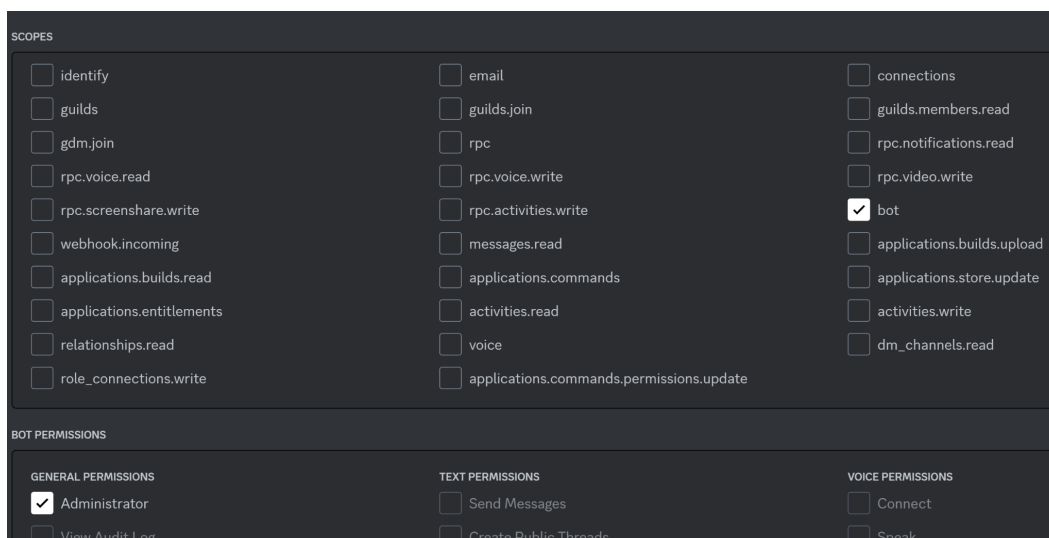
Una vez creado, nos vamos a la pestaña “Bot”, y pulsamos en crear



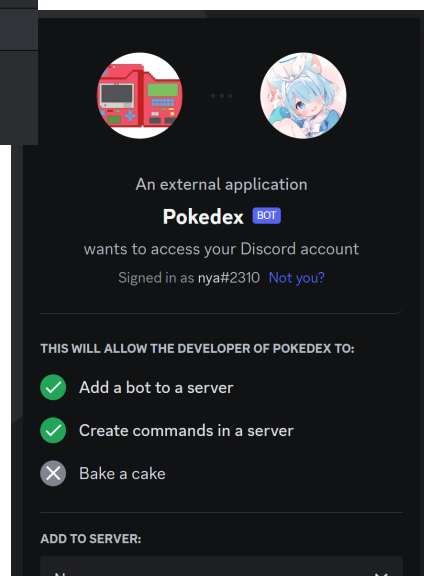
Una vez hecho eso, le daremos permisos al bot, si queremos asegurarnos de que todo funciona correctamente, para asegurarnos le damos todos los derechos, o el derecho de administrador



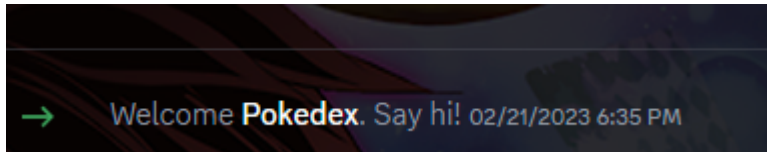
Y seguidamente en “URL Generator”, generamos el enlace con los mismos permisos



Una vez hecho esto, veremos que en la parte inferior de la ventana, ha aparecido un enlace, solo tendremos que copiarlo y pegarlo en nuestro navegador.



Seleccionamos el servidor al que queramos añadir, y pulsamos en continuar



4. Creación del proyecto python

Para crear el proyecto tenemos dos alternativas, instalarnos python en nuestra máquina que actuará de servidor, o crearlo en un servicio online el cual se encargará de mantener activo al bot sin necesidad de que tengamos nuestro equipo encendido

Yo usaré la opción en local, pero el siguiente enlace es un buen sitio en caso de querer usar la segunda opción

[Replit: the collaborative browser based IDE - Replit](#)

4.1. Primeros pasos

Lo más básico antes de empezar, es importar las librerías que necesitaremos, en este caso siendo discord la más importante

Después deberemos crear una variable que contenga nuestro TOKEN, o identificador de nuestro bot

```
TOKEN = 'MTA3NzY0MzU2NzY0NTg2ODExNA.Gt0P05.o1CixBGJDHde6m9_6OrCWj5x0iB61h-HMTFOOU'
```

(Discord tiene una colaboración estricta con github y detectará si un token se ha hecho público. Como se requiere que el proyecto sea un repositorio público, el TOKEN nunca estará actualizado en el repositorio)

Este segundo paso se debería de poder hacer desde la página en la que creamos el bot, pero para asegurarnos de que nuestro bot tenga todos los permisos necesarios para funcionar sin problemas, añadiremos la siguiente línea

Y el último paso antes de tenerlo todo listo, crearemos un objeto Bot para poder ejecutar nuestro bot

```
client = commands.Bot(command_prefix='.', intents=intents)
```

Command_prefix: Es el carácter que tendremos que escribir al inicio de nuestros mensajes para ejecutar los comandos que crearemos. Es recomendable que sea un carácter poco usado tanto al inicio de los mensajes, como por otros bots para evitar problemas.

Algunos malos ejemplos de prefijos son: "A", "@", "\$", "/", "#" (es muy posible que @ no funcione ya que se usa para las menciones)

Intents: Es lo que ya he mencionado antes, es aquí donde lo usaremos

Y ya solo quedaría añadir esta línea al final de nuestro código

```
client.run(TOKEN)
```

4.2. Crear un evento

Crearemos un evento cuando queramos que el bot haga algo cuando pase algo en específico en el servidor.

Por ejemplo el `on_ready()`, este evento se activará en el segundo en el que el bot pase a estar online

Otro muy usado es el `on_message(message)`. Este capturará cada mensaje que se mande en el servidor. Es IMPORTANTE, que si usamos este evento tendremos que añadir la siguiente línea para que nuestros comandos sigan funcionando

```
@client.event
async def on_message(message):
    await client.process_commands(message)
```

Después de la línea del `await` podremos añadir cualquier cosa que deseemos hacer

Pero hay muchos otros eventos que podremos gestionar:

- `on_message_edit(before, after):`
- `on_message_delete(message):`
- `on_member_join(member):`
- `on_member_update(before, after):`
- `on_reaction_add(reaction, user):`
- `on_reaction_remove(reaction, user):`
- `on_raw_reaction_add(payload):`
- `on_raw_reaction_remove(payload):`
- `on_channel_create(channel):` Triggered when a new channel is created in a server.
- `on_channel_delete(channel):` Triggered when a channel is deleted in a server.
- `on_guild_join(guild):` Triggered when the bot joins a new server.
- `on_guild_remove(guild):` Triggered when the bot is removed from a server.

4.3. Crear un comando

A diferencia que en los eventos, las posibilidades para los comandos es infinita, podremos crear tantos como queramos o necesitemos, con el nombre que queramos

Para poder usar un comando que creemos, el mensaje en discord tiene que tener la siguiente sintaxis: `<prefijo><nombreDelMetodo> <parametros (en el caso de que haya)>`

A continuación veremos dos ejemplos:

Sin parametros

.info

```
@client.command()
async def info(ctx):
    des = ""
    Comandos de TestBot\n
```

Con parametros

.pokemonName Pikachu

```
@client.command()
async def pokemonName(ctx, name):
    channel = client.get_channel(ctx.channel.id)

    http = urllib3.PoolManager()
```

Como podemos apreciar “ctx” se repite en ambos casos, en discord ctx es el objeto que contiene toda la información de un mensaje, quien lo ha mandado, a donde, cuando, que contiene etc..

Este objeto nos permitirá que el bot envíe mensajes únicamente a donde debe, y no a todos los canales (chats) que pueda

5.Creación de la base de datos

Como ya he indicado antes, la base de datos será creada en phpMyAdmin abriendo los puertos mediante XAMPP

5.1. Tablas

En esta parte voy a enumerar las tablas explicando el uso que tendrá en la aplicación

5.1.1. Pokemon

Es la tabla más importante ya que contiene la información de todos los Pokémon que se usarán en la base de datos, los cuales en total son 890 especies distintas

La información de cada pokemon sera:

- Numero
- Nombre
- Descripción
- Tipos (por tabla muchos a muchos)

- HP
- Ataque
- Defensa
- Velocidad
- Ataque Especial
- Defensa Especial
- Region

5.1.2. Types

En esta tabla se almacenarán todos los tipos existentes de Pokémons. De cada tipo almacenaremos únicamente el nombre

5.1.3. Poke_Type

Aquí es donde le damos uso a la tabla Types, relacionado con la tabla Pokemon.

Almacenaremos qué tipos pertenecen a cada Pokemon.

Esta relación es una tabla diferente debido a que un Pokémon puede tener hasta 2 tipos y varios Pokémons pueden tener el mismo tipo, lo que hace que esto sea una relación de muchos a muchos

5.1.4. Region

Al igual que en la tabla Types, aquí solo almacenaremos el nombre

5.1.5. Users

Aquí guardaremos los datos identificativos de cada usuario para que pueda hacer uso de la operación más importante, la cual explicaré con la siguiente tabla. La tabla consta de la siguiente información

- Id
- Username
- discriminator

En discord cada nombre de usuario consta de dos partes. El nombre en sí, y luego el **discriminator**, el cual es número aleatorio de cuatro cifras gracias al cual es posible que muchos usuarios tengan el mismo nombre, eliminando así el posible descontento de usuarios que no puedan usar su nombre debido a que alguien ya lo tiene.

Un nombre de usuario de discord se ve así: nya#2310

En una reciente actualización de discord se van a eliminar los discriminators, es por esto que los he eliminado del proyecto

5.1.6. Caught

Esta tabla es en parte la esencia de la Pokédex, que cada usuario pueda llevar un registro de todas las especies de Pokémon que ha atrapado.

Relacionándose con las tablas de Users y Pokémon, podremos llevar un registro de captura para cada usuario

5.1.7. Type_Effectiveness

Esta tabla sirve para darle una funcionalidad muy útil a la Pokédex, funcionalidad que no tienen las Pokédex de los juegos.

Esta te permitirá saber cuán efectivo es un ataque de un tipo hacia un pokemon. La estructura será la siguiente

- offensiveType
- defenisveType
- effectiveness

5.2. Introducción de los datos básicos

La pokédex es una base de datos con una grandísima base de datos, que de estar completa al 100% consta de más de 1000 especies diferentes de pokémon. Sin contar todas sus subtablas que se tendrían que crear para poder gestionar la información de las habilidades, huevos, naturalezas etc

Debido a que este módulo no se centra en el uso de bases de datos, se ha intentado simplificar la introducción de los datos básicos (pokemons y sus tipos) lo máximo posible

Primero pensé que debería haber algún fichero de base de datos que me permitiese importar una base de datos ya creada con toda esta información, pero esto no fue posible. Pero mientras buscaba el archivo de base de datos me percate de que habían archivos con la información que buscaba. Archivo que podría usar para insertar los datos a través de Insomnia

Cuando ya tenía uno que me podía servir, vi que tenía varios atributos que no eran realmente útiles, como el nombre en Kanji japonés. Es por esto que investigue alguna forma de eliminar y cambiar el nombre a los atributos de forma sencilla, lo que me permitió preparar completamente el archivo para su uso

6. Funciones bot

En este punto voy a explicar las diferentes funciones implementadas en el bot para uso de todos los usuarios. La gran mayoría las puede hacer cualquier usuario, pero hay algunas que tendrán un requisitos, los cuales explicare junto con el comando en sí

6.1. Buscar por número de pokédex

Sintaxis: .pokemon <numero>

Este comando buscará el pokemon identificado por el número escrito, y lo mostrará con todos sus datos

6.2. Buscar por nombre

Sintaxis: `.pokeName <nombre>`

Este comando buscará un pokemon con el nombre que le hayamos indicado. Se debe escribir correctamente respetando mayúsculas y cualquier puntuación que este pueda tener

Ejemplo: `.pokeName Farfetch'd`

Un ejemplo de mal uso seria de la siguiente forma: `.pokeName farfetchd`

6.3. Buscar según la región

Este comando buscará a todos los pokémon de la región indicada y los mostrará uno a uno por medio de la interacción del usuario.

El mensaje que contiene el pokemon vendrá con 4 reacciones de flechas, las cuales tendrán diferentes funciones:

“<<”: Nos enviará al inicio de la lista (En kanto aparecería Bulbasaur)

“<”: Nos mostrará el pokémon anterior al anterior (En Charizard sería Charmeleon)

“>”: Nos mostrará el pokémon anterior al anterior (En Charizard sería Charmeleon)

“>>”: Nos enviará al finalde la lista (En kanto aparecería Mew)

6.4. Registrarse

Sintaxis: `.register`

Este comando guardará nuestro nombre de usuario en la base de datos, el cual en las próximas semanas será único sin necesidad del discriminator, como ya he explicado antes

Este comando es el requisito para poder hacer algunos comandos en los que se necesitan datos acerca del usuario.

6.5. Capturar pokemon

Sintaxis: `.catch <numero>`

Este comando registrara como capturado al pokémon que indiquemos en el comando. Para poder usarlo es necesario haber ejecutado el comando `“.register”`. Ya que la captura se registrará en relación a nuestro nombre de usuario

6.6. Ver progreso total

Sintaxis: `.caughtCount`

Este comando nos informará de nuestro progreso actual en la pokedex, nos dirá cuantos pokemons hemos capturado sobre todos los que hay “7 de 809”

6.7. Ver progreso en una región

Sintaxis: `.caughtRegion <nombreRegion>`

Este comando al igual que el anterior nos indicará nuestro progreso, pero solo acerca de la región que indiquemos.

Por lo que si preguntamos sobre kanto, el resultado sería: 3 de 151

6.8. Ver pokemons restantes en total

En complemento con el ver la cantidad que llevas capturados, este mostrará los que te faltan por capturar. Te mostrará toda la información disponible de cada uno de los pokemons que te falten por capturar

El funcionamiento es el mismo que en `pokeRegion`, podrás desplazarte a través de los diferentes pokemons gracias a 4 reacciones

6.9. Ver pokemons restantes en una región

6.10. Consultar efectividad

Sintaxis: `.effectiveness <NombreTipo> <NombrePokemon>`

En los juegos de antes los cuales no estaban pensados para niños recién nacidos, era muy típico toparse con un pokémon del cual no tenías ni idea de que podías hacerle por que su combinación de tipos era muy peculiar, ejemplos de esto pueden ser Spiritomb y Volcanion

Pero tampoco hacía falta que saliese un pokemon raro para tener este problema, puesto que aprenderse las debilidades de los 18 tipos existentes no es moco de pavo

Esta funcionalidad añadida es la que pone solución a este problema, te permitirá ver cuál de tus ataques tendrá una mayor efectividad ante tu rival

Se requiere el nombre para el uso ya saberse el número de pokedex de tu rival no es la cosa más común

Ejemplo: `.effectiveness Electric Wingull`

Respuesta: “The Electric type has a x4.0 effectiveness against Wingull

6.11. TypeChart

Sintaxis: `.typeChart`

En caso de que quieras un resultado más rápido pero menos preciso, podrás usar este comando para ver todas las debilidades de todos los tipos

Por el momento solo muestra una imagen, pero en una versión más avanzada se trabajara con los datos de la base de datos

6.12. Whos that pokemon

Sin contar a Ash y Pikachu, es fácilmente lo más famoso y reconocido de todo pokemon. En la serie animada se te mostraba la silueta de un Pokémon y tras una pausa (normalmente ponían anuncios ahí) se te revelaba la identidad del pokemon

Al ser algo tan icónico, y relacionado con la información tratada en la pokédex. He decidido que sería interesante hacer una función. Su uso es un poco más complejo que los anteriores

La sintaxis es `.play`

Pero la cosa no acaba ahí, ya que tras ver la imagen tendremos un total de 5 segundos para escribir nuestra respuesta, la cual deberá seguir la siguiente sintaxis

Sintaxis respuesta: `guess<Nombre>`

6.13. Info

Sintaxis : `.info`

Y por último pero no por ello lo menos importante, el comando de información. Este comando explicará de forma concisa cómo se utiliza cada uno de los comandos de la aplicación.

Gracias a este comando la sintaxis y uso de todos los comandos será accesible de forma muy sencilla para todos los usuarios

7. Recomendaciones para el año que viene

Durante el desarrollo del proyecto he ido pensando en lo que vendría bien para añadir al temario del año que viene.

Este proyecto ha tocado ámbitos vistos en Acceso a Datos e Interfaces, por lo que sólo se indicarán mejoras para dichos módulos

7.1. Acceso a datos

Para mejorar este módulo, añadiría una actividad para aprender a hacer relaciones muchos a muchos “@ManyToMany” en JPA

Aunque no es algo complicado, no es tan sencillo como las dos vistas durante el año y creo que son lo suficientemente habituales como para que merezca la pena hacer una actividad para enseñar como se hace

7.2. Interfaces

Cambiaría el tiempo que se usa para las interfaces de móviles para hacer otras cosas, ya sea más tiempo para c++ o para python. Ambos lenguajes son bastante buenos y debido a que ya se le hace mucho hincapié a las interfaces de móviles en PMDM, no veo necesario gastar tiempo en ello en este módulo

Como segundo punto, creo que una aplicación del estilo de este proyecto es muy ejemplo para enseñar programación asíncrona en python ya que es muy visible la forma en la que el por ejemplo el bot espera a la información antes de enviar el mensaje

7.3. Extra

Algo que he notado durante las prácticas es que en clase se nos enseña a hacer el programa y ya. Pero de normal la cosa no acaba ahí, ya que después el programa se debe de poder ejecutar en cualquier máquina sin que haya ninguna complicación

Es por eso que en cada lenguaje utilizado se nos debería enseñar como hacer un ejecutable que funcione en una alta mayoría de equipos. Tal y como se hace con las APK de android en PMDM

Unos ejemplos de ellos serían como hacer un jar que no de problemas con las dependencias al llevarnoslo a una máquina distinta o como traspasar todas las librerías externas que estamos usando en un proyecto python

8. Conclusiones

Me ha parecido un proyecto bastante entretenido y he aprendido bastantes cosas haciéndolo.

También he cogido una gran soltura en python, lo que me podrá ser de gran ayuda en el futuro

9. Bibliografia

[Welcome to discord.py \(discordpy.readthedocs.io\)](https://discordpy.readthedocs.io)
[JavaScript Arrays \(w3schools.com\)](https://www.w3schools.com/js/js_arrays.asp)
[Many-To-Many Relationship in JPA | Baeldung](#)
[Pokemon Discord Bots | The #1 Discord Bot List \(top.gg\)](#)
[Discord | Tu sitio para hablar y pasar el rato](#)
[python - Discord.py @bot.event - Stack Overflow](#)

10. Recursos utilizados

Discord
Python
Java
Maven
SpringBoot
SpringBoot Initializer
Practicas acceso a datos
Github
Xampp
Discord.py
OBS
Adobe Premiere