 

**PROGRAMACION BASICA**

**OSVALDO HABIB GONZÁLEZ GONZÁLEZ**

**PROYECTO FINAL PIA**

**SEBASTIAN CALDERON CARRILLO 2087472**

**DIEGO ERNESTO CAMARILLO SALAZAR 1996387**

**HECTOR ALAN HERNANDEZ GONZALEZ**

**JOEL DE JESUS GALVAN CAMPOS 2159634**

**LEONARDO ISAAC VELA CORTES**

**LUIS ANTONIO ROCHA ESCOBEDO**

**Planteamiento del problema:**

**Que fenómeno o situación a analizar.**

Se busca analizar las diferentes estadísticas de los pokémons base. Estadísticas como ataque, defensa, vida, velocidad, altura y peso.

**Porque es relevante**

Existen personas que dedican mucho tiempo a competir en cosas relacionadas con Pokémon, por lo que les y nos es de ayuda para analizar cuales serian nuestras mejores opciones para utilizar contra algún rival.

**Datos que se necesitan obtener**

La API que hemos escogido cuenta con un sinfín de datos, pero en nuestro caso los que más nos son útiles son los datos numéricos, en los cuales se incluyen ataque, defensa, vida, velocidad, altura y peso. Estos datos intuitivamente van relacionados al nombre de cada pokémon, siendo este su identificador.

**Descripción de la API elegida**

La API que escogimos es publica y gratuita. Esta API se creo con la idea de unificar el sitio de donde las demás páginas de análisis Pokémon obtenían su información, volviéndose ya casi un standard en cuanto a datos actualizados se trata. La API tiene una restricción de peticiones HTTPS por hora, la cual puede variar dependiendo del tráfico, pero en general se recomienda usar sabiamente y con cautela la cantidad de peticiones solicitadas. Es por eso por lo que optamos por crear un cache con la información obtenida de la API, y solamente hacemos requests HTTPS cuando necesitemos actualizar la información.

**Estructura de datos**

Dependiendo de las distintas funciones que hicimos, es que se creo una estructura de datos adecuada para la necesidad a resolver. Aunque, también podemos decir que la estructura “principal” es la que ya nos ofrece la API en sus respuestas, la cual vendría siendo un diccionario iterable con elementos key:value, el key siempre es un string describiendo que es lo que contiene su correspondiente value. El value puede cambiar dependiendo de que es la información que contenga. A veces son listas de diccionarios, listas de tuplas, a veces solo es un integer, etc.

A continuación, describiremos las estructuras de datos utilizadas en las funciones correspondientes:

**Justificación del tratamiento de datos aplicado**

**Minutas de trabajo**

**Minuta 2 de mayo de 2025**

**Roles asignados:**

• **Líder:** Héctor

• **Programador**: Sebastián y Héctor

• **Diseñador de algoritmo**: Joel y leonardo

• **Encargado del diagrama**: Luis y Diego

**Objetivo para el día:**

Ponernos de acuerdo en cuanto a la idea del proyecto a realizar, junto con la organización de cada miembro para cumplir los roles requeridos

**Acuerdos tomados:**

Los roles asignados, la persona que supervisará el trabajo y las actividades de cada uno.

**Dificultades encontradas:**

Organización para las actividades a realizarse dado que se inició antes de vacaciones.

**Próximos pasos:**

• Realizar el diagrama de flujo y diseño del algoritmo.

• Realizar la codificación en Python.

**Cuadro comparativo de APIs exploradas**

**Algoritmo**

**Diagrama de flujo**

**Resumen del proyecto con hallazgos relevantes**

**Guion del podcast**

**Capturas de pantalla y elementos visuales de apoyo**