**Bloque 1 — Tuplas (15–20 min)**

1. **Ficha inmutable**  
   Crea una tupla pokemon\_base con: (nombre, tipo\_principal, nivel\_inicial)

* Comprueba acceso por índice y **desempaquetado**.
* Intenta “cambiar” el nivel dentro de la tupla (debería fallar) y explica por qué en un print explicativo.

1. **Tupla como clave**  
   Crea un diccionario registro\_capturas donde la **clave** sea la tupla (ruta, hora) y el **valor** sea el nombre del Pokémon.

* Añade 3 capturas.
* Pregunta si existe una clave concreta usando in.

1. **Ranking Compacto**  
   Crea una lista de **tuplas** ranking = [(nombre, nivel), ...].

* Recorre la lista y muestra “#1 NOMBRE — Nivel X”, etc.
* Encuentra la tupla con mayor nivel sin usar funciones avanzadas (solo bucles y condicionales).

**Bloque 2 — Diccionarios (25–30 min)**

1. **Pokédex mínima**  
   Crea pokedex = { "Sprigatito": {"tipo": "Planta", "nivel": 7}, ... } con **3 entradas**.

* Añade un cuarto Pokémon.
* Sube el nivel de uno en +2.
* Muestra todos en formato “Nombre (Tipo) — Nivel”.

1. **Medias por zonas**  
   avistamientos = { "Zona Sur": [7, 5, 3], "Zona Este": [4, 4, 6] } (niveles vistos)

* Calcula la **media** de cada zona y muéstrala (usa / len(lista) o / 3 si fijas 3 valores).
* Di qué zona tiene la media más alta.

1. **Diccionario de listas + búsqueda**  
   En movimientos = { "Sprigatito": ["Arañazo", "Hoja Afilada"], ... }:

* Añade un movimiento a dos pokémon.
* Pide por teclado un **nombre** y muestra sus movimientos o el mensaje “No encontrado”.

**Bloque 3 — Funciones (30–40 min)**

1. **calcular\_media\_niveles(lista\_niveles)**  
   Devuelve la media (float).

* Prueba con [5, 10, 7] y con [10, 10, 10].
* Muestra el resultado con un print claro (sin :.2f si no quieres).

1. **subir\_nivel(pokedex, nombre, puntos)**  
   Recibe el diccionario del ejercicio 4, un nombre y puntos a subir.

* Si existe, actualiza; si no, **no** hagas nada pero informa por pantalla.
* Devuelve True/False según haya actualizado.

1. **mejor\_pokemon(pokedex)**  
   Devuelve el **nombre** con mayor nivel (si hay empate, el primero que encuentre).

* Recorre con .items() y variables auxiliares (mejor\_nombre, mejor\_nivel = 0).

**Bloque 4 — Clases y objetos (45–60 min)**

1. **Clase Pokemon**  
   Atributos: nombre (str), tipo (str), nivel (int), movimientos (lista/tupla).  
   Métodos:

* subir\_nivel(puntos) → aumenta nivel (mínimo 0).
* aprender\_movimiento(mov) → añade si no está y si hay hueco (p.ej., máximo 4).
* resumen() → devuelve texto: "NOMBRE (TIPO) — Nivel X — Movs: ...".  
  Crea 3 instancias y prueba los métodos.

1. **Clase Equipo**  
   Atributos: miembros (lista de Pokemon, máx. 6).  
   Métodos:

* agregar(pokemon) → añade si hay espacio y no está repetido por nombre.
* quitar(nombre) → elimina por nombre si existe.
* promedio\_nivel() → media de niveles.
* mostrar\_equipo() → imprime cada resumen().  
  Crea un equipo, añade 3–4 pokémon, sube niveles a alguno y muestra el equipo.

**Mini-proyecto integrador (60–90 min)**

**“Gestor de Equipo Pokémon (modo texto)”**  
Un **programa pequeño** que use **tuplas + diccionarios + funciones + clases**.

**Requisitos mínimos**

* Usa tu clase Pokemon y la clase Equipo del Bloque 4.
* **Datos semilla**: crea 3 pokémon iniciales (puedes guardarlos en un **diccionario** plantilla\_inicial con claves por nombre y valores con una **tupla** de atributos básicos, p.ej. (tipo, nivel, movimientos\_iniciales) para construirlos).
* **Menú en bucle** (while True con opción de salida) con opciones:
  1. **Añadir Pokémon** al equipo (desde la plantilla\_inicial por nombre o creando uno nuevo con input).
  2. **Quitar Pokémon** por nombre.
  3. **Subir nivel** de un miembro (pide nombre y puntos).
  4. **Aprender movimiento** (pide nombre y movimiento).
  5. **Ver equipo** (usa mostrar\_equipo() del equipo).
  6. **Ver promedio de nivel** (función o método).
  7. **Buscar el de mayor nivel** (función aparte que recorra equipo.miembros y devuelva el nombre).
  8. **Salir**.
* **Condicionales**: valida que no se supere el tamaño 6, que el nombre exista, etc.
* **Bucles**: para recorrer miembros y para mantener el menú.
* **Nada de comprehensions ni excepciones** (mantén lo visto).

**Criterios de aceptación**

* El menú no crashea ante entradas simples (si pones una letra cuando esperas un número, basta con mostrar un aviso y seguir; no hace falta capturar excepciones).
* mostrar\_equipo() lista todos con el formato de resumen().
* promedio\_nivel() devuelve un número coherente con los niveles actuales.
* “Mayor nivel” devuelve uno de los presentes; si el equipo está vacío, muestra un mensaje claro.
* No hay duplicados por **nombre** en el equipo.

**Ideas de pruebas rápidas (tú mismo)**

1. Añade 3 pokémon (uno debe ser tu Sprigatito ❤️).
2. Sube 2 niveles a uno y 5 a otro; confirma el **promedio**.
3. Haz aprender un movimiento nuevo a uno que tenga menos de 4.
4. Quita uno y comprueba que se actualiza la lista.
5. Muestra el “mayor nivel” y verifica que coincide.