

Patrones de diseño 1

STRATEGY PATTERN

STRATEGY PATTERN

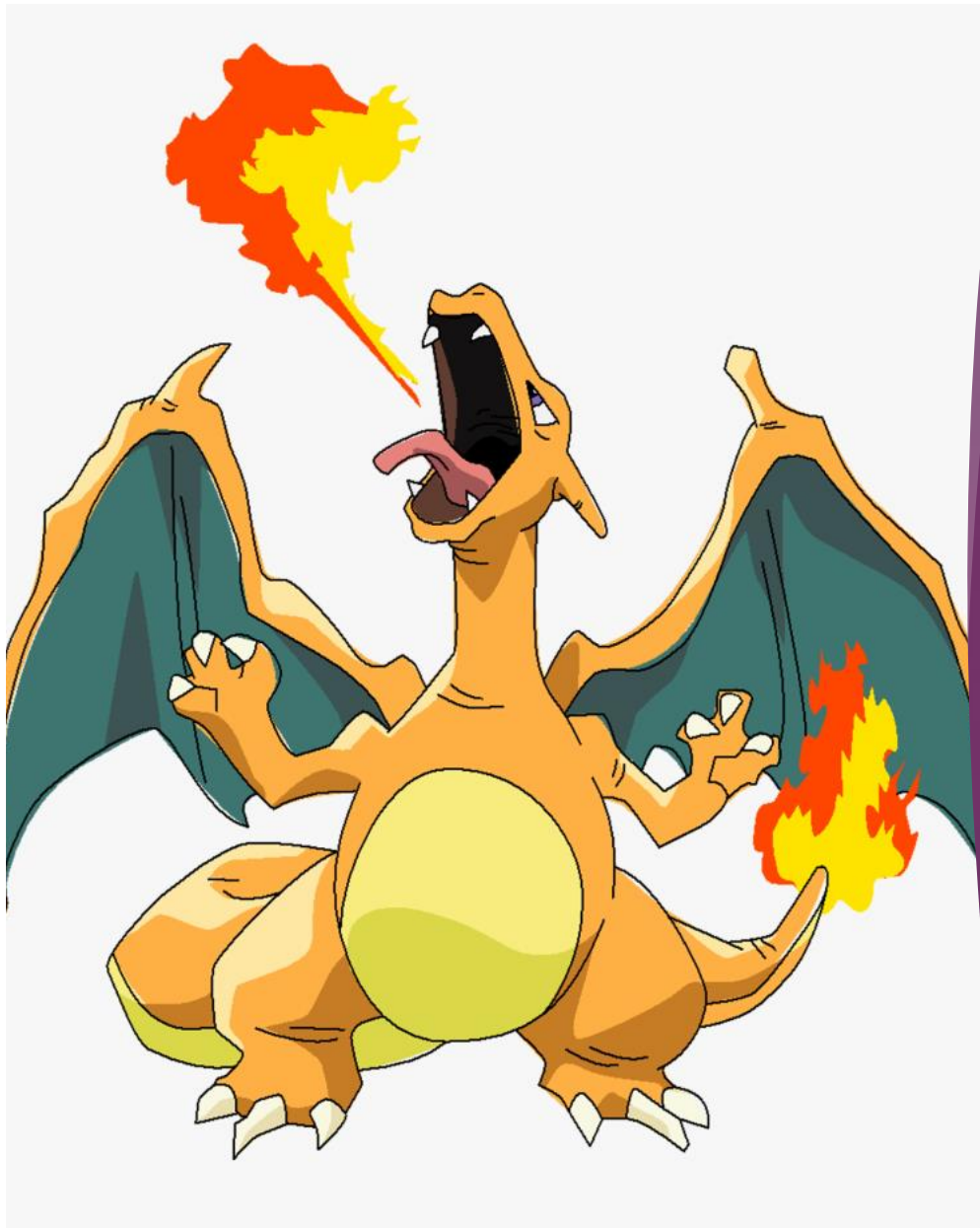
OBJECT - BEHAVIORAL

► Intención:

- Definir una familia de algoritmos, que tienen una funcionalidad similar que difiere en implementación. Para luego poder hacerlos intercambiables.

► Motivación:

- Hay situaciones donde hay ciertos algoritmos que hacen intrínsecamente una misma acción pero que se implementan diferente. En estos casos es bueno aislar esos algoritmos así podemos intercambiarlos cuando se nos plazca.



Ataques Pokémon



Ataques Pokémon

```
class Pokemon {  
    performAttack(attackType: string, gameState: GameState): GameState {  
        if (attackType == "hiperrayo") {  
            gameState.enemy.hp = gameState.enemy.hp - 20;  
        } else if (attackType == "lanzallamas") {  
            gameState.enemy.hp = gameState.enemy.hp - 10;  
            gameState.enemy.state = "Tu pokemon esta quemado";  
            setTimeout(() => {  
                gameState.enemy.hp = gameState.enemy.hp - 10;  
                gameState.enemy.state = "";  
            }, 1000);  
        } else if (attackType == "danza lluvia") {  
            gameState.weather = "empezo a llover ☁☁☁";  
        }  
        return gameState;  
    }  
}
```

Ataques Pokémon


A

Nombre ↕	Tipo ↕	Categoría ↕	Concurso ↕	Inglés ↕	Japonés ↕	Romanizado ↕
A bocajarro	LUCHA		Dureza ¹	Close Combat	インファイト	Infaito
A defender	BICHO		Ingenio	Defend Order	ぼうぎょしれい	Bougyo shirei
Abatidoras	PLANTA		Ingenio ²	Worry Seed	なやみのタネ	Nayami no tane
Absorbefuerza	PLANTA			Strength sap	ちからをすいとる	Chikara o suitoru
Absorber	PLANTA		Ingenio	Absorb	すいとる	Suitoru
Ácido	VENENO		Ingenio	Acid	ようかいえぎ	Youkaieki
Acoso	BICHO		Dulzura	Infestation	まわりつく	Matowaritsuku
Acróbata	VOLADOR		Carisma	Acrobatics	アクロバット	Akurobatto
Acua aro	AGUA		Belleza	Aqua Ring	アクアリング	Akuaringu
Acua cola	AGUA		Belleza ³	Aqua Tail	アクアテール	Akuateru
Acua jet	AGUA		Carisma ²	Aqua Jet	アクアジェット	Akuajetto
Acupresión	NORMAL		Dureza ⁴	Acupressure	つぼをつく	Tsubowotsuku
Adaptación	NORMAL		Belleza	Nature Power	しぜんのちから	Shizen no chikara
Aerochorro	VOLADOR		Carisma	Aeroblast	エアロブラスト	Aeroblast
Afilagarras	SINIESTRO		Dulzura	Hone Claws	つめとぎ	Tsumetogi
Afilar	NORMAL		Dulzura	Sharpen	かくばる	Kakubaru
Agarre	NORMAL		Dureza	Vice Grip	はさむ	Hasamu
Agarrón	NORMAL		Dureza	Crush Grip	にぎりつぶす	Nigiri tsubusu
Agilidad	PSÍQUICO		Carisma	Agility	こうそくいどう	Kōsoku idō
Agua lodosa	AGUA		Dureza	Muddy Water	だくりゅう	Dakuryuu
Aguante	NORMAL		Dureza	Endure	こらえる	Koraeru
Aguijón letal	BICHO		Carisma	Fell Stinger	とどめばり	Todome-bari
Agujero negro aniquilador	SINIESTRO			Black Hole Eclipse	ワールズエンドフ ォール	Burakku Hōru Ikuripusuu
Aguzar	NORMAL			Laser focus	とぎすます	Togisumasu
Aire afilado	VOLADOR		Carisma	Air Cutter	エアカッター	Air cutter
Al ataque	BICHO		Ingenio	Attack Order	こうげきしれい	Kougeki shirei
Ala de acero	ACERO		Carisma	Steel Wing	はがねのつばさ	Hagane no tsubasa
Ala mortífera	VOLADOR		Carisma	Oblivion Wing	デスウイング	Desu Uingu
Alarido	SINIESTRO		Dureza	Snarl	パークアウト	Baakuauto
Alboroto	NORMAL		Dulzura	Uproar	さわぐ	Sawagu
Aligerar	ACERO		Belleza	Autotomize	ボディパーズ	Bodipaaji
Alivio	NORMAL		Dulzura	Refresh	リフレッシュ	Refresh
Alud	HIELO		Belleza ⁴	Avalanche	ゆきなだれ	Yuki nadare

Ataques Pokémon



Codifiquemos los Ataques usando Strategy Pattern



```
export default interface IAttack {  
  name: string;  
  executeAttack(gameState:GameState): GameState;  
}
```

Interfaz Para los ataques


```
export default class Terremoto implements IAttack {
    name: string = "Terremoto";


    executeAttack(gameState: GameState): GameState {
        gameState.enemy.hp = gameState.enemy.hp - 20;
        gameState.weather = "El suelo tiembla";

        return gameState;
    }
}
```

```
export default class HiperRayo implements IAttack {
    name: string = "Hiper Rayo";

    executeAttack(gameState: GameState): GameState {
        gameState.enemy.hp = gameState.enemy.hp - 20;
        return gameState;
    }
}
```

Clases de ataque



```
export default class Pokemon {  
  
  private attack: IAttack;  
  
  setAttack(attack: IAttack) {  
    this.attack = attack;  
  }  
  
  performAttack(gameState: GameState): GameState {  
    console.log("Pokemon ataca con " + this.attack.name);  
    return this.attack.executeAttack(gameState);  
  }  
  
}
```

Clase Pokemon

Strategy Pattern en código

```
export default class Pokemon {  
  private attack: IAttack;  
  
  setAttack(attack: IAttack) {...}  
  
  performAttack(gameState: GameState): GameState {  
    ...this.attack.executeAttack(gameState);  
  }  
}
```

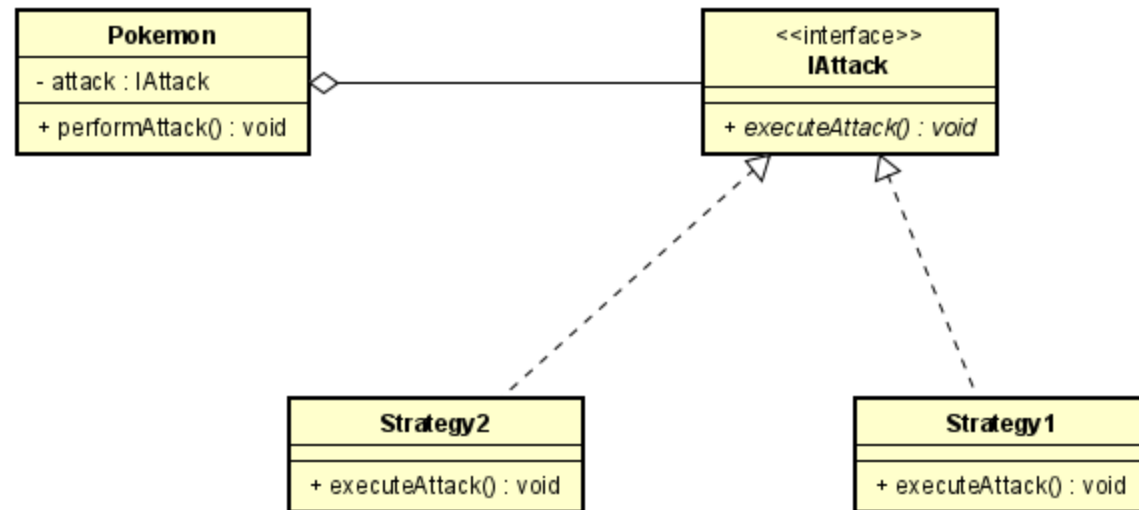
```
export default interface IAttack {  
  name: string;  
  executeAttack(gameState: GameState): GameState;  
}
```

```
export default class HiperRayo implements IAttack {  
  name: string = "Hiper Rayo";  
  
  executeAttack(gameState: GameState): GameState {  
    ...  
  }  
}
```

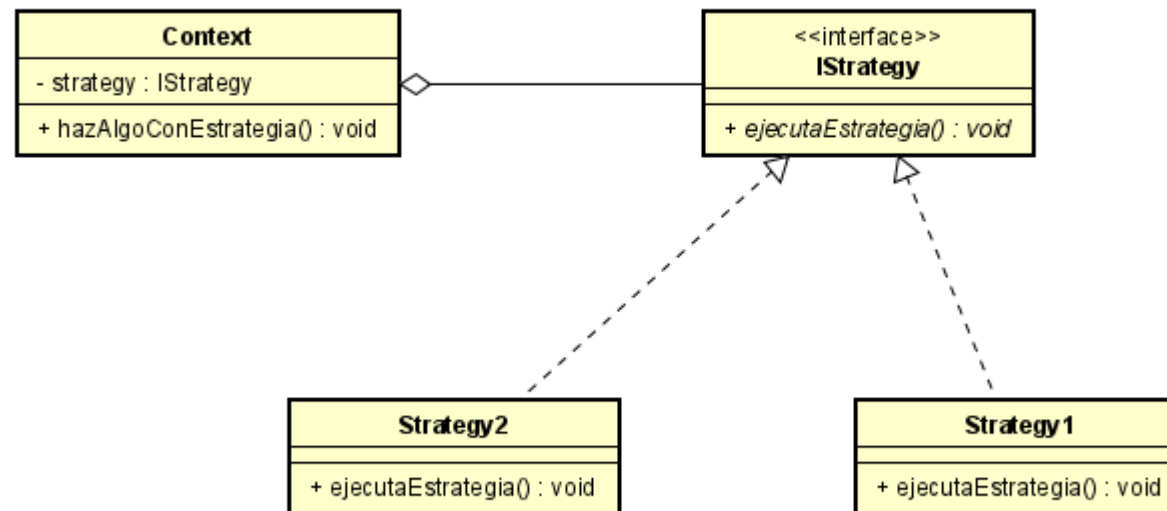
```
export default class Terremoto implements IAttack {  
  name: string = "Terremoto";  
  
  executeAttack(gameState: GameState): GameState {  
    ...  
  }  
}
```



El patron Strategy



El patron Strategy



Categorías de patrones

Objects – Classes



Class



Object

Strategy

Categorías de patrones

Creational – Behavioral - Structural

