

1. Interfaz IEffect

- Define un contrato para todos los efectos del juego.
- Método obligatorio: `Execute(Vector2 position, GameObject owner = null)`.
- Permite ejecutar efectos de forma uniforme.

2. Interfaz IPersistentEffect

- Hereda de IEffect.
- Métodos:
 - `ApplyTo(GameObject player)` → aplica el efecto de forma permanente.
 - `RemoveFrom(GameObject player)` → opcional, permite remover efectos.
- Se usa para mejoras que duren toda la run.

3. ScriptableObjects de Efectos

ProjectileEffect

- Abstracto, hereda IEffect.
- Método `Execute` obligatorio.

ExplosionEffect (ejemplo)

- Hereda ProjectileEffect.
- Ejecuta daño, knockback y VFX en un área.
- Ignora al owner si está configurado.

4. PickupEffectItem

- Trigger cuando el jugador toca el pickup.
- Pausa el juego mientras muestra nombres de efectos.
- Activa un efecto final mediante RunEffectManager.
- Destroy del pickup al finalizar.

5. EffectSpawner

- Lista de efectos (`ScriptableObject`).
- Método `TriggerEffects(Vector2 position, GameObject owner = null)` ejecuta todos los efectos.

6. RunEffectManager

- Singleton que controla efectos activos durante la run.
- Métodos:
 - `ActivateEffect(ScriptableObject effect)`
 - `IsEffectActive(ScriptableObject effect)`

- `GetActiveEffects()`
- Guarda efectos persistentes o temporales en un `HashSet`.

7. Efectos persistentes

- Implementan `IPersistentEffect`.
- Se aplican al jugador usando `ApplyTo()`.
- Ejemplo de uso: incrementar stats, añadir VFX que siga al jugador.
- Flujo:

Pickup recogido → `IPersistentEffect?` → `ApplyTo(player)` → `RunEffectManager.ActivateEffect(effect)`

8. Ejemplo de Aura Persistente

```
public class AuraEffect : ScriptableObject, IPersistentEffect
{
    public GameObject auraPrefab;
    private GameObject instantiatedAura;

    public void ApplyTo(GameObject player)
    {
        if (auraPrefab == null || instantiatedAura != null) return;
        instantiatedAura = Object.Instantiate(auraPrefab, player.transform);
        instantiatedAura.transform.localPosition = Vector3.zero;
        DontDestroyOnLoad(instantiatedAura);
    }

    public void RemoveFrom(GameObject player)
    {
        if (instantiatedAura != null) Object.Destroy(instantiatedAura);
    }

    public void Execute(Vector2 position, GameObject owner = null) {}
}
```

- El objeto instanciado sigue al jugador durante toda la run.

Resumen Visual del Flujo:

```
[Jugador toca Pickup] → [PickupEffectItem OnTriggerEnter] → [Mostrar nombres de efectos] → [Activar efecto]
|
└─ Si es persistente → ApplyTo(player) →
```

```
RunEffectManager.ActivateEffect(effect)  
  └ Si es temporal → Execute(position, owner)
```