# Unity Vector

**Vector2**

**Vector3**

Representa las 3D.

Se declara de la siguiente manera: Vector3(x, y, z).

Propiedades:

back: Forma abreviada de Vector3(0, 0, -1).

[down](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-down.html): Forma abreviada de Vector3(0, -1, 0).

[forward](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-forward.html): Forma abreviada de Vector3(0, 0, 1).

[left](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-left.html): Forma abreviada de Vector3(-1, 0, 0).

[one](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-one.html): Forma abreviada de Vector3(1, 1, 1).

[right](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-right.html): Forma abreviada de Vector3(1, 0, 0).

[up](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-up.html): Forma abreviada de Vector3(0, 1, 0).

[zero](https://docs.unity3d.com/2018.3/Documentation/ScriptReference/Vector3-zero.html): Forma abreviada de Vector3(0, 0, 0).

normalized: Conserva la dirección del Vector pero la longitud la convierte en 1.

magnitude: retorna la longitud del vector.

* Lerp: Sirve para calcular la interpolación linear. La interpolación linear sirve para calcular cualquier punto en una recta. Por ejemplo si mi A=(x=1, y=5) y mi B=(x=3, y=15) si intento calcular x=2 la y seria 15\*2 = 30, 30/3= 10. Y=10

En unity con esto en el update se creara un efecto de movimiento del objeto hacia la posición objetivo de forma gradual.

Vector3.Lerp(posición\_actual, posición\_objetivo, float t)

T: si es 0 devuelva la posición de origen, si es 1 la posición objetivo, si es 0,5 devuelve la interpolación linear. En la práctica dará un efecto de mayor o menor suavidad, cuanto más cerca de 0 más suave y cuando más cerca de 1 mas rápido.

Vector3.Lerp(0, 100, 0.25f)

Lo que hace es, en el primer update el valor sería el 25% de la diferencia entre el valor actual y el objetivo (100-0=100), (25% de 100 = 25), si actualizamos el valor inicial a 25 en el siguiente update el valor que dara sería el 25% de la diferencia. (100-25=75, 25% de 75 = 16,5). Si actualizamos el valor dara 25+16,5=41,5. Haciendo esto sucesivamente provocara ese efecto de suavidad al llegar al objetivo. Lo único malo es que el objeto que lo lleva nunca descansa, siempre está moviéndose.

Vector3.Dot(VectorA, VectorB): Devuelve si los vectores son perpendiculares unos de otros(0), si van en la dirección opuesta(-1) o en la misma dirección(1).

Vector3.Cross(VectorA, VectorB): Devuelve un nuevo Vector con la posición perpendicular entre los dos que le hemos pasado.