

# Dragon Ball Z

[Contexto General](#)

[Descripción del Dominio](#)

[Guerreros](#)

[Semillas de Ermitaño](#)

[Alianzas](#)

[Trajes](#)

[Dinámica de peleas](#)

[Tipos de Guerreros](#)

[Puntos de Ataque](#)

[Poderes](#)

[Esferas del Dragón](#)

[Requerimientos](#)

[Aclaraciones y sugerencias](#)

[Pensar mucho en objetos y mensajes](#)

[Quién conoce a quién](#)

[Ir resolviéndolo de a poco, partir el trabajo](#)

[Buenos y malos](#)

[Asuencia de traje](#)

[Creación de instancias](#)

[Recomendaciones para el desarrollo del TP](#)

[Extras](#)

[Ataque inteligente](#)

[Super Sayajin](#)

[Esferas únicas](#)

# Contexto General

Capsule Corp. es una empresa de grandes proyectos tecnológicos situada en un mundo fantástico donde siempre hay un malo que se quiere apoderar del mundo o simplemente demostrar que es el más fuerte.

Algo muy particular de este mundo fantástico, es que existen 7 "esferas del Dragón" que reunidas pueden servir para invocar al dragón Shen Long y pedirle un deseo. Luego de concedido el deseo, Shen Long reparte las esferas por el mundo y para pedir otro deseo habrá que volver a reunir las.

Bulma, la hija de los fundadores de Capsule Corp., es amiga de Goku y un grupo de fuertes guerreros. Estos guerreros usarán su fuerza para reunir las esferas del dragón, ganar dinero en Torneos de Artes Marciales, o incluso salvar al mundo de los malos que quieran apoderarse de él o de las esferas.

Para darles una mano, Bulma nos pide que le ayudemos a Capsule Corp. a hacer algunos programas para ayudar a sus amigos a estar bien prevenidos.

En concreto, tenemos que construir un sistema de simulación de pelea entre guerreros, contemplando la idea de que un grupo de guerreros aliados logre obtener las esferas del dragón quitándoselas a otros guerreros.

Para simular una pelea hay que tener en cuenta varios factores.

# Descripción del Dominio

## Guerreros

Los guerreros serán los involucrados en las peleas, en las que participarán haciendo diferentes ataques a un adversario (ver luego en sección de Ataques). Todos los guerreros cuentan con un nivel de pelea que se corresponde a cuán fuertes son (por ejemplo, Goku tiene un nivel de pelea de 100.000, y es más fuerte que Krilin que tiene un nivel de pelea de 28.000).

Los guerreros también cuentan con una cantidad de energía y de puntos de vida. Cada guerrero nace con esos 1000 puntos de vida, que es el máximo de puntos de vida que puede tener en cualquier momento; y 200 puntos de energía, la energía de un guerrero puede variar sin topes.

Si un guerrero tiene 50 puntos de vida o menos se lo considera moribundo, y si llega a 0 puntos ese guerrero murió y queda fuera de combate. Se puede revivir a un guerrero, para eso habrá que juntar las 7 Esferas del Dragón y pedirle el deseo a Shen Long (ver sección de *Esferas del Dragón*).

La energía es lo que les servirá para atacar a otros guerreros y se irá consumiendo cuando el guerrero hace ataques.

## Semillas de Ermitaño

Si en algún momento un guerrero vivo queda con poca energía (menos de 200), puede utilizar unas semillas especiales llamadas "Semillas de Ermitaño" que se usan para recuperar energía y vida.

Cada guerrero pueden tener una cantidad determinada de Semillas de Ermitaño. Cuando un guerrero se come una semilla de Ermitaño, recupera toda su vida, y además recupera tanta energía como el 5% de su nivel de pelea + una cantidad extra según el tipo de guerrero. Obviamente, además, pasa a tener una Semilla de Ermitaño menos en su haber.

Si un guerrero quiere comer una Semilla de Ermitaño para recuperarse y no tiene ninguna, le puede pedir una a uno de sus aliados. Los guerreros pueden ser de uno de dos bandos: "buenos" o "malos". Los guerreros buenos siempre le dan una Semilla de Ermitaño al aliado que se la pide. Los malos sólo ceden una Semilla de Ermitaño a un aliado si no es la última que les queda.

## Alianzas

Los guerreros pueden tener muchos aliados. Las condiciones para que un guerrero sea aliado de otro se especifican según el tipo de guerrero más adelante.

La relación de aliado es bidireccional, por lo tanto:

- si Goku incorpora como aliado a Vegeta, eso implica que automáticamente Vegeta incorpore como aliado a Goku.
- por lo tanto, para que esta relación se dé, deben cumplirse las condiciones en ambos sentidos: Goku debe aceptar a Vegeta como aliado y viceversa.

Las alianzas no se rompen; por lo tanto, no hace falta contemplar validaciones sobre cómo quedan las alianzas una vez que se rompe una.

## Trajes

Los guerreros pueden ponerse un traje. Es decir, pueden tener uno, ninguno, pero no dos trajes puestos, y pueden cambiarlo.

Todos los trajes sirven de protección; cuando un guerrero recibe un ataque (ver sección siguiente), el traje resiste el 10% de los puntos de ataque recibidos. Existen trajes especiales, que además de resistir este 10%, resisten 20 puntos extra.

Hay otro tipo de traje, de entrenamiento. Los trajes de entrenamiento son pesados, y el guerrero que lo lleva puesto gasta energía al recibir ataques ya que cargar con el traje le presenta un trabajo adicional. Cuando un guerrero que está usando un traje de entrenamiento recibe un ataque, además de la pérdida en puntos de vida, pierde también 20 puntos de energía; pero cada 3 ataques que recibe, gana 100 puntos de nivel de pelea.

## Dinámica de peleas

Cuando un guerrero ataca a otro, lo hace con una cantidad de *puntos de ataque* (ver sección *Puntos de Ataque*), que representan la intensidad del ataque realizado.

El guerrero atacado recibe esos puntos de ataque, aunque su traje resistirá una parte de esos puntos, y de lo que pase el traje, el guerrero puede resistir a su vez una parte. Los puntos de ataque que no resiste ni el traje ni el guerrero, el atacado los pierde en puntos de vida. P.ej. me atacan con 1000 puntos de ataque, mi traje resiste 100, yo resisto 200; la consecuencia es que pierdo 700 puntos de vida.

No es condición para que haya un ataque que el atacante y el atacado deban ser de

distinto bando; un malo puede atacar a un malo o a un bueno, lo mismo un bueno. Sí hay que respetar las alianzas: un guerrero sólo puede atacar a otro guerrero si atacante y atacado no son aliados.

Resumiendo: en una pelea, según el tipo de guerrero que reciba el ataque y el tipo de traje que lleve puesto, los puntos de ataque recibidos se traducirán a una cierta cantidad de puntos de vida perdidos, que se desviarán de la relación  $\text{puntos de ataque recibidos} = \text{puntos de vida perdidos por la resistencia que ofrecen el traje y el guerrero}$ . Incluso puede darse el caso donde entre la resistencia propia del guerrero y la resistencia del traje, el ataque recibido quede anulado y no se decremente la vida del guerrero atacado.

Por ejemplo: Goku le hace un ataque a Freezer de 100 puntos de ataque, Freezer en ese momento tiene un traje normal que resiste 10 puntos, de los 90 que quedan Freezer resiste 20 por ser un guerrero monstruo (ver más abajo); por lo tanto sólo se ve afectado en perder 70 puntos de vida. Inicialmente Freezer tenía 760 puntos de vida, y después del golpe de Goku quedó con 690.

Otro ejemplo: Krilin ataca a Raditz con 60 puntos de ataque; Raditz en ese momento tiene un traje especial que le retiene 26 puntos, a Raditz le llegan 34, por ser saiyajin (ver más abajo) y tener 2000 puntos de energía resiste  $(40+3.4) = 43.4$  puntos, que son más que los que pasaron el traje; por lo tanto, Raditz no recibió puntos de daño y sus puntos de vida no se vieron afectados.

## Tipos de Guerreros

Existen varios tipos de guerreros:

- **Terrícolas:** Sólo pueden hacer ataques que les consuman menos de 500 puntos de energía. Cuando comen una Semilla de Ermitaño ganan 200 puntos de energía extra. Todos los guerreros terrícolas son buenos y sólo pueden aliarse con buenos. Un guerrero terrícola se puede aliar hasta con 5 guerreros con nivel de pelea menor a 15.000, más todos los que quiera de nivel de pelea de 15.000 o más. Al ser atacados resisten el 5% de los puntos de ataque que deje pasar el traje que tengan en ese momento.
- **Monstruos:** Todos los monstruos son malos, pueden aliarse con guerreros de cualquiera de los 2 bandos, sin condiciones. Cuando comen una Semilla de Ermitaño no ganan puntos extras. Al ser atacados resisten 20 puntos fijos independientemente de los puntos de ataque recibidos, que son los que deja pasar el traje que tengan en ese momento.
- **Saiyajin:** Cuando comen una Semilla de Ermitaño ganan como puntos de energía extra  $100 * \text{cantidad de aliados que tengan}$ . Depende de cada saiyajin si es malo o bueno, y puede aliarse con cualquiera de los dos bandos mientras el nuevo bando contrario tenga candidatos que totalicen un nivel de pelea mayor (les gustan los desafíos). Al ser atacados resisten el 10% de los puntos de ataque recibidos + el 2% de la energía que les queda; de lo que deje pasar su traje.

## Puntos de Ataque

Está claro que en una pelea se ven involucrados: el guerrero atacante, el guerrero atacado, y el ataque que está usando.

La intensidad de un ataque (la cantidad de *puntos de ataque*) que produce un ataque depende del tipo de ataque, y en varios casos de la energía que el guerrero que lo haya realizado le aportó al mismo, incluso en un caso también contribuyen algunos de los aliados del guerrero atacante, que también pierden la energía aportada.

La cantidad de puntos de ataque que produce un poder al realizarse, y sus consecuencias sobre el atacante y eventualmente sus aliados (básicamente pérdida de energía, pero en algún caso más cosas), dependen del tipo de poder (ver sección siguiente).

## Poderes

Los poderes pueden ser:

- Livianos: Son poderes que no sacan energía al guerrero que ataca, pero no pueden hacerse si un guerrero tiene menos de 50 puntos de energía. Cada poder liviano produce siempre la misma cantidad de puntos de ataque, que puede variar entre los diferentes poderes livianos.

Por ejemplo: Goku puede realizar un “disco de energía” que produce 100 puntos de ataque porque lo tiene aprendido y tiene 300 puntos de energía, pero ni Krilin ni Vegeta pueden realizar este mismo poder; Krilin porque tiene sólo 40 puntos de energía (esto le impide realizar cualquier poder liviano), Vegeta porque no lo tiene aprendido. Luego de que Goku realice el “disco de energía” seguirá teniendo 300 puntos de energía.

- Potentes: Le sacan al guerrero el 25% de su energía, y sus puntos de ataque respetan la siguiente fórmula:  $(\text{nivel de pelea del guerrero} / 500 * \text{potencia del ataque})$ . La potencia del ataque es una característica configurable de cada poder potente.

Por ejemplo: El poder potente “Kame Hame Ha” tiene 5 de potencia de ataque. Goku, que tiene 100.000 de nivel de pelea y 500 de energía, tiene aprendido el poder. El “Kame Hame Ha” hecho por Goku produciría  $100.000 / 500 * 5 = 1000$  puntos de ataque, y Goku quedaría con 375 puntos de energía.

- Colectivos: Le “piden” energía a aliados del guerrero que esté haciendo el ataque. Le sacan una cierta cantidad de puntos de energía a cada aliado especificado para cada poder colectivo y otros 200 puntos de energía al atacante mismo, y los puntos de ataque que producen son iguales a la cantidad total de energía recaudada. Para que un guerrero pueda realizar este poder, tiene que tener al

menos esos 200 puntos de energía y también una cantidad mínima de aliados vivos (especificada para cada poder colectivo) que puedan darle la cantidad de puntos de energía que le sacaría a cada uno.

Por ejemplo: el poder “Genkidama” requiere tener al menos 5 aliados vivos que puedan entregar energía, y le saca 50 puntos a cada aliado que pueda entregar 50 puntos (es decir, si tiene sólo 50 puntos de energía, se los saca todos; si sólo tiene 40 lo pasa por alto y no le saca nada). Goku quiere hacer el poder colectivo “Genkidama” para atacar a Freezer, y de los 10 aliados vivos que tiene sólo 8 pueden entregarle 50 puntos de energía. Puede hacer el poder porque tiene al menos 5 aliados vivos que le pueden entregar la energía requerida por la “Genkidama” y él tiene 500 puntos de energía. Los puntos de ataque producidos por este poder serán  $200 + 8 * 50 = 600$ . Freezer recibirá 600 puntos de ataque. Goku pierde 200 puntos de energía, y cada uno de los aliados que contribuyó pierde 50.

- Vengativos: Todos los poderes vengativos sacan la misma cantidad de energía (que debe ser configurable), y los puntos de ataque son correspondientes a  $250 * \text{la cantidad de aliados del atacante que estén muertos}$ . Para poder realizar un poder vengativo, un guerrero debe tener la energía suficiente y además deben haber muerto una cierta cantidad mínima de sus aliados, especificada para cada poder vengativo; además, el atacante debe tener al menos un aliado vivo.

Por ejemplo: Vegeta puede hacer el poder “Final Flash”, que como cualquier poder vengativo saca (por ejemplo) 150 puntos de energía. Vegeta puede realizar el poder “Final Flash” ya que tiene 1000 puntos de energía, le quedan 150 puntos de vida, tiene aliados vivos, y además 4 de sus aliados están muertos y para realizar el “Final Flash” al menos 2 de sus aliados debieron haber muerto. Los puntos de ataque producidos por el “Final Flash” si fuera realizado por Vegeta serían  $4 * 250 = 1000$ , y Vegeta perdería  $1000 / 100 * 20 = 200$  puntos de vida, por lo tanto muere después de realizar el poder.

## Esferas del Dragón

Las esferas del dragón son 7, y cada esfera tiene una cantidad de estrellas de 1 a 7 (entonces está la esfera de 1 estrella, la de 2 estrellas, etc.). El sistema no deberá permitir la creación de esferas “truchas” que tengan otra cantidad de estrellas, ni la modificación de esta cantidad una vez creadas. Estas operaciones deben fallar.

Para simplificar el modelo, no se necesitan validaciones respecto a esferas duplicadas y se asume que no se encontrarán en el mundo dos esferas válidas de la misma cantidad de estrellas (por ejemplo, se asume que no ocurriría la situación donde dos guerreros tienen cada uno una esfera de 2 estrellas, cuando debería existir una sola).

Se tiene en cuenta que todas las esferas ya están poseídas por algún guerrero. La

invocación a Shen Long la hace un guerrero cuando quiere, pero para que un guerrero pueda invocar a Shen Long, entre él y sus aliados deben tener reunidas las 7 esferas. Para reunir las 7 esferas, los guerreros deben sacárselas a otros que no son sus aliados. Para que un guerrero pueda sacarle la esfera a otro, debe luchar contra él. Si el guerrero al que atacó queda moribundo o directamente muere, el guerrero atacante toma la/s esfera/s que tuviera el atacado.

Si un guerrero llegase a morir tras realizar un poder vengativo (que sacan vida al que los hace) y tiene en su posesión una o más esferas, se las entrega todas a uno cualquiera de sus aliados vivos, haya muerto el contrincante o no; si además el contrincante muere o queda moribundo, sus esferas pasan también al mismo aliado vivo del atacante que se sacrificó por la causa.

Cuando un guerrero invoca a Shen Long y tiene reunidas todas las esferas entre él y sus aliados, el dragón le concede el deseo y se reparten las esferas nuevamente entre los guerreros, de la siguiente forma:

- Si tiene 7 o más aliados vivos, le da cada una a 7 aliados cualquiera (no importa a cuales, ni tiene porqué ser aleatorio).
- En caso contrario, y si tiene suficientes enemigos vivos, reparte las que puede entre sus aliados vivos, y el resto entre enemigos vivos cualquiera.
- Si aún así sigue sin haber esferas sin repartir, estas quedan en manos del guerrero que las tiene actualmente.

Los guerreros buenos, cuando reúnen las 7 esferas del dragón, usan el deseo para revivir a uno de sus aliados caídos; si tienen más de un aliado caído, eligen al que tiene mayor nivel de pelea. Los guerreros malos usan el deseo para volverse un 50% más fuertes ellos mismos (aumentar el nivel de pelea) y un 10% más fuertes a todos sus aliados.

Por ejemplo: Goku, que es bueno y aliado de Vegeta, invoca a Shen Long, lo puede hacer porque Goku tiene algunas de las esferas y Vegeta las restantes, y además tiene sentido la invocación, o sea, tiene algún aliado muerto.

Lo que va a pasar en la invocación es revivir a alguno de los aliados muertos de Goku. Además de esto, se reparten las esferas.

Supongamos que Vegeta atacó a tres guerreros en este orden: Freezer, Cell, y Krilin. En este caso, las esferas que tiene Vegeta pasan a Krilin, si Krilin estuviera muerto o moribundo a Cell, si Cell también estuviera muerto o moribundo a Freezer, si los tres estuvieran muerto o moribundo entonces a un aliado cualquiera de Vegeta (Goku u otro) que no esté ni muerto ni moribundo, si tampoco hay ningún aliado que pueda recibir las esferas, se las queda Vegeta.

## Requerimientos



Se pide que el programa cumpla con las siguientes funcionalidades, respetando el dominio detallado anteriormente.

1. Alianzas

- a. Saber si un guerrero A puede ser aliado de otro B; o sea si A acepta a B como aliado y viceversa.

Tener en cuenta que no es correcto que un guerrero se alíe con alguien de quien ya es aliado ni consigo mismo, además de las condiciones detalladas en el dominio.

- b. Aliar a un guerrero con otro, si se puede llevar a cabo la alianza. Si no se puede, que falle con una excepción. Recordar que las alianzas son siempre bidireccionales.

2. Semillas

- a. Saber si un guerrero puede comer una Semilla de Ermitaño.
- b. Que un guerrero se coma una Semilla de Ermitaño, si puede. Si no puede, que falle con una excepción.

3. Poderes y niveles de ataque

- a. Hacer que un guerrero aprenda un poder
- b. Saber para un guerrero cuántos puntos de vida le quitaría un ataque de  $n$  puntos de ataque que representan la intensidad del ataque que recibiría, donde se indica el " $n$ ". Son los puntos que no resiste el traje y tampoco resiste el guerrero.  
P.ej. "Goku, si te atacaran con 1000 puntos de ataque, ¿cuántos puntos de vida perderías?"
- c. Hacer que un guerrero reciba una cantidad de puntos de ataque; al recibir esta cantidad el guerrero deberá decrementar su vida.  
Tener en cuenta también lo que pasa si el traje que tiene el guerrero es de entrenamiento (pérdida de energía, y eventualmente aumentar su nivel de pelea).
- d. Saber cuánta energía necesitaría usar un poder si lo realizara un guerrero en particular. P.ej. "¿Cuánta energía necesitaría el Kame Hame Ha si lo realiza Goku?"
- e. Saber cuántos puntos de ataque produciría un poder si lo realizara un guerrero en particular. P.ej. "¿Cuántos puntos de ataque produciría el Kame Hame Ha si lo realiza Goku?"
- f. Saber si un guerrero puede realizar un poder, según las condiciones indicadas al describir los poderes.
- g. Que un guerrero realice un poder en particular para atacar a otro guerrero, si puede hacerlo. En este caso se indica explícitamente qué poder usar, p.ej "Goku, atacá a Vegeta usando kameHameHa". Si no puede hacerlo, que falle con una excepción.

Hay que incluir el daño al atacado y las consecuencias sobre el atacante que se indican para cada tipo de poder.

Si el ataque deja moribundo o muerto al contrincante con esferas, las toma; también tener en cuenta qué pasa con las esferas que tuviera el

atacante si muere en el ataque (poderes vengativos).

- h. Saber para un guerrero, qué poderes podrían hacer él y/o al menos uno de sus aliados.

#### 4. Ataque selectivo

- a. Pedirle a un guerrero que ataque a otro guerrero, usando uno cualquiera de sus poderes aprendidos que pueda usar en ese momento.

En este caso, al igual que en los siguientes ítems de este punto, no se especifica qué poder usar, eso lo decide el atacante.

- b. Pedirle a un guerrero que ataque a otro guerrero, usando el poder que más daño haría al contrincante dentro de los poderes que tiene aprendidos y que puede usar.
- c. Pedirle a un guerrero que ataque a otro guerrero, usando de entre los poderes que tiene aprendidos y que puede usar, alguno (cualquiera) que produzca al menos X puntos de ataque, donde el X se especifica en cada caso.

P.ej. "Goku, atacá a Piccolo con al menos 1324 puntos de ataque".

Si no puede, que elija el que más daño le haría al contrincante.

#### 5. Resurrecciones

- a. Pedirle a un guerrero el conjunto de aliados que necesitan ser revividos (o sea, que estén muertos, claro).

#### 6. Llamada a Shen Long

- a. Saber si un guerrero junta todas las esferas del dragón entre él y sus aliados. Pista: Acá conviene que haya alguien que conozca todas las esferas posta para saber si las tenemos todas; no es correcto comparar por cantidad de esferas porque mañana nos mudamos a otro planeta con otras esferas como Namek y nos cambian tanto la cantidad de esferas como la condición de si son posta.
- b. Saber si un guerrero puede llamar a Shen Long. Ese guerrero lo puede llamar si junta todas las esferas del dragón entre él y sus aliados, si no está muerto ni moribundo, y si tiene sentido que pida un deseo. P.ej. siempre tiene sentido que Freezer (que es malo) pida un deseo porque los malos le piden a Shen Long ser más fuertes, pero no tiene sentido que Goku (que es bueno) pida un deseo si no tiene al menos un aliado para revivir.
- c. Saber, para un guerrero, a qué otro guerrero debería cederle sus esferas en la repartija de esferas que ocurre ante cada llamada a Shen Long (las reglas del último atacado etc. que se describen en el apartado sobre esferas del dragón); si no se las puede dar a nadie, el elegido es él mismo (se la queda).
- d. Que un guerrero llame a Shen Long y pida su deseo. Si no puede hacerlo, que falle con una excepción. Recordar que luego de pedido un deseo, las esferas se reparten entre distintos guerreros según lo detallado en el dominio.



# Aclaraciones y sugerencias

## Pensar mucho en objetos y mensajes

Para resolver cada punto del TP van a necesitar, además de lo que pide el punto, otras cosas.

El TP les va a salir mucho mejor si hacen el esfuerzo de aclararse a ustedes mismos cuáles son esas cosas, y para cada una que detecten, eligen a qué objeto se la pido, con qué mensaje, con qué parámetro/s (si los necesita). Van a ver que si se concentran en estas decisiones, en lugar de en acomodar código, el código les va a quedar más chico.

P.ej. para saber si un guerrero puede realizar un poder, y para realizarlo, en el caso de los poderes vengativos y colectivos hay que manejar los aliados vivos y los aliados muertos del guerrero en cuestión. Entonces, ¿qué hago?. Pensemos con los aliados vivos.

Antes de PDP, tal vez hubieran pensado cómo recorrer la colección, algo así:

```
vivos := Set new.
unGuerrero aliados do:
    [:aliado | (aliado puntosDeVida > 0) ifTrue: [vivos add: aliado]].
"en vivos tengo el conjunto de los aliados vivos de unGuerrero"
```

ahora que aprendimos colecciones, ya sé que le puedo delegar la selección a una colección

```
vivos := unGuerrero aliados select: [:aliado | aliado puntosDeVida > 0]
```

pero ... desde el punto de vista de objetos hay algo que no está muy bien. Cuando pongo

```
aliado puntosDeVida > 0
```

¿qué estoy averiguando? Si el aliado está vivo. Y ... ¿quién sabe si un guerrero está vivo? ¡Claro, el guerrero! Entonces

```
vivos := unGuerrero aliados select: [:aliado | aliado estaVivo]
```

acá ya estamos mucho mejor, pero podemos dar un paso más ... ¿qué era lo que quería? El conjunto de aliados vivos de un guerrero. Bien:

```
vivos := unGuerrero aliadosVivos.
```

y ahora si que el código dice lo que hace: obtiene los aliados vivos de unGuerrero.

Para llegar directamente a la última versión, una forma es preguntarme:

- ¿qué es lo que quiero? el conjunto de aliados vivos de unGuerrero
- ¿quién tiene sentido que sepa resolver eso? el guerrero, claro.
- ¿qué mensaje le envío? aliadosVivos
- ¿con qué parámetro/s? No necesito

Una vez que decidí esto:

- el código de aliadosVivos es muy sencillo; sería la versión anterior, si me avivo

que lo que quiero saber de c/u es si está vivo o no. Son 6 palabras.

- el código que usa aliadosVivos queda muy sencillo, con 2 palabras resuelve la cuestión.

Otro ejemplo: tengo que hacer que un guerrero pierda n puntos de energía, ¿le digo

```
unGuerrero energia: unGuerrero energia
```

?

Noooo! Lo que quiero decirle es que pierda n de energía, entonces

```
unGuerrero perdeEnergia: n
```

Y otro más: en varios lugares necesito al conjunto de un guerrero más sus aliados (p.ej. cuando tengo que saber si entre él y sus aliados tienen todas las esferas); bueno, hago que el guerrero entienda el mensaje conAliados.

Eso va a hacer que tal vez algunos de sus objetos tengan un vocabulario bastante extenso, entiendan un montón de mensajes. OK, más adelante (en otras materias) discutirán si eso es o no un problema, y técnicas para manejarlo.

De esta forma, no deberían quedarles métodos de más de 4 ó 5 líneas, con mayoría de 1 ó 2.

Si escriben un método de 10 líneas, es muy-muy probable que estén pensando algo en forma incorrecta desde el punto de vista de objetos, que es el que tienen que practicar en este TP.

## Quién conoce a quién

En algunos casos van a necesitar que A conozca a B y también B conozca a A; fíjense de establecer las relaciones necesarias para que siempre que un objeto deba enviarle un mensaje a otro lo pueda acceder, y no más.

## Ir resolviéndolo de a poco, partir el trabajo

Fíjense que hay varios tipos de guerreros y de poderes. Entonces p.ej. se puede, en el punto 1, primero resolverlo para terrícolas y monstruos, y recién cuando anda agregarle saiyajines. Y en el punto 3, se puede hacer todo p.ej. para poderes livianos y potentes que son más sencillos, y después agregar colectivos y vengativos.

Fíjense que una vez que se tiene el punto 3 andando para algunos poderes, se puede seguir con los puntos 4 al 9, y después si agrego los otros poderes en forma adecuada, no debería ser necesario tocar nada de lo que hice.

## Buenos y malos

Busquen una solución en la que las cosas en las que buenos y malos se portan distinto (cesión de la semilla de Ermitaño, deseo pedido a Shen Long) no se resuelva con poner en el código

```
self esBueno
  ifTrue: [lo que tengo que hacer si soy bueno]
  ifFalse: [lo que tengo que hacer si soy malo]
```

sino que se delegue en otros objetos, que representan a los bandos.

## Ausencia de traje

Puede ocurrir que un guerrero no tenga un traje. Una forma de encararlo es preguntar todo el tiempo si el traje está o no (con un `traje ifNil:` por ejemplo).

Pero la idea es resolverlo de una forma “objetosa”, sin recurrir a eso: pensar en el “no traje” como objeto.

## Algoritmos complejos

Hay en el dominio algunos algoritmos que parecen complejos, pero si se emplean colecciones adecuadamente, se simplifican enormemente haciendo innecesario el uso de if's y do's.

## Creación de instancias

Verán que el modelo es complejo, por lo tanto, tienen que armar muchos casos de prueba, pero muchos.

Es importantísimo que acompañen el código con un workspace con muchos ejemplos, y bien organizado. Tengan en cuenta que armar el workspace es una tarea no menor dentro del desarrollo del TP, o sea, calculen que les va a llevar un tiempo. Si no se organizan para construir los objetos de ejemplo, se les va a complicar. El uso de constructores ayuda bastante a simplificar esta tarea. Por ejemplo, en lugar de:

```
dosKilosDePapa = Artículo new.
dosKilosDePapa peso: 20.
dosKilosDePapa peligroso: false.

envio38 := Envio new.
envio38 personaQueRetira: 'Juan Ramirez'.
envio38 agregarArticulo: dosKilosDePapa.
```

se puede hacer

```
envio38 := Envio queRetira: 'Juan Ramirez'.
envio38 agregarArticulo: (Articulo noPeligrosoDePeso: 20).
```

y así de seis líneas de workspace pasamos a dos.

## Recomendaciones para el desarrollo del TP

Este TP es complejo y no se puede encarar en solitario, y sin prestar mucha atención a los requerimientos. Dejamos dos sugerencias para mejorar el desarrollo del mismo:

- Compartir el código utilizando una herramienta de control de versiones: compartir código por mail es engorroso y propenso a errores y pérdidas de código. Por eso existen herramientas de control de versiones como Monticello; [acá te dejamos un tutorial de la cátedra](#).
- Usar un Workspace es útil, pero muchas veces tedioso de ejecutar: hay que correr la “prueba” a mano, cada vez que se modifica algo. Existen herramientas que automatizan la prueba del código: los así llamados frameworks de testeo unitario, como SUnit; [también hicimos un tutorial sobre eso!](#)

¿Es tema de la materia? No. ¿Es obligatorio usarlo? Tampoco. Pero les traerá grandes beneficios, a fin de año nos cuentan ;)

## Extras

Si querés el 10, pueden hacer algunos de estos puntos extra:

### Ataque inteligente

Ahora se requiere poder decirle a un guerrero que ataque a otro, sin indicarle el criterio que deberá seguir para elegir con qué poder aprendido atacar. El criterio se debe poder configurar para cada guerrero, y cuando al guerrero se le pida atacar deberá elegir el poder siempre bajo ese criterio que le configuramos.

Por ejemplo: En lugar de pedirle a Goku que ataque con el poder que menos energía le consume o pedirle que ataque con cualquier poder, configurarlo para que siempre que se le pida atacar con un poder, elija el poder que más daño le haría al adversario, y simplemente pedirle que ataque a un adversario. Después si le cambio el criterio a "ahorrá energía", y le vuelvo a pedir que ataque, en lugar de elegir el que más lastima, elige el que menos energía le cuesta.

## Super Saiyajin

Incluir la posibilidad de que un Saiyajin se transforme en Super Saiyajin, lo puede hacer cuando decida, pero debe tener más de 1000 puntos de energía y no estar moribundo. Pueden volver a ser un Saiyajin normal a voluntad, o forzosamente si después de recibir un ataque queda moribundo.

Cuando un Saiyajin está transformado en Super Saiyajin, sus ataques son el 30% más fuertes y resisten a los ataques recibidos el 60% más.

## Esferas únicas

Se desea que el sistema garantice que no haya esferas duplicadas. Es decir, que no se pueda crear dos veces esferas con la misma cantidad de estrellas.