Resumen pauta E2

IMPORTANTE ❗

- Tal como se indica en la pauta, los descuentos por cada error son sugeridos, es decir, queda a criterio del ayudante corrector si hacer un descuento mayor o menor. Por ejemplo:
  ▶️ Una función con 3 niveles de indentación podría tener un menor descuento que una función con 7 niveles.
  ▶️ Una variable llamada `a` podría tener mayor descuento que un mal nombre pero que sea más descriptivo y que demuestre un esfuerzo.
- Este resumen aplica para el semestre 2025-2. Si estás revisando esto en otro semestre, considera que la pauta pudo haber cambiado.

Cap 2

Nombre Descriptivo

Descuento máximo por tema: `ilimitado`

> Los nombres dados a archivos, clases, métodos, atributos y variables describen la completamente la funcionalidad de lo que representan.

Descuentos sugeridos por error:

`-0.05` Nombres poco descriptivos.

`-0.05` No se hacen distinciones significativas.

`-0.5` Por método de 100 líneas o más.

> No, no nos equivocamos en poner este último descuento acá y no en el capítulo

Nombre de Clase y Variable

Descuento máximo por tema: `0.5`

> Los nombres de clases y variables deben incluir un sustantivo que describa lo que representan.

Descuentos sugeridos por error:

`-0.05` Por clase sin sustantivo.

`-0.05` Por variable sin sustantivo.

No encapsular constantes

> 3.
>
> Cuando nos encontramos con un método tan grande, el nombre del método será muy general para todos los detalles que se encuentran en él, y por lo tanto, el nombre del método está mal, pero no se arregla poniendo otro nombre, sino que se arregla disminuyendo el largo del método.
>
> Por otro lado, es muy probable que este método cuente con variables del tipo `player1`; estos nombes no son malos cuando nos entramos en el contexto adecuado (método pequeño), pero cuando estamos en un método tan largo, estos nombres hay que tratarlos como si fueran atributos de clase (ya que se encuentran en un contexto muy extenso) y, por lo tanto, son malos
>
> **(*)** Ahora bien, toda regla tiene su excepción. En este caso, una excepción es que el método sea parte de una Factory. El método que crea objetos tenderá a ser una cadena interminable de ifs (o similar). No descontar en esos casos.

Nombre de Método

Descuento máximo por tema: `0.5`

> Los nombres de los métodos deben incluir un verbo que describa su

Descuento máximo por tema: `0.5`

> Se debe evitar el uso de números, strings o bools directamente en el código.

Descuentos sugeridos por error:

`-0.05` Por usar nombres de un carácter o números directo en el código.

Una palabra por concepto

Descuento máximo por tema: `0.5`

> Está bien usar nombres como `fetch`, `retrieve` y `get`, o `manager`, `driver` y `controller`, pero se debe elegir solo uno y no usarlos todos al mismo tiempo como equivalentes.

Descuentos sugeridos por error:

`-0.05` Por violación del principio.

> funcionalidad.

Descuentos sugeridos por error:

`-0.05` Por método sin verbo.

Cap 3

Condiciones Punto Base

- El alumno no tiene ningún método de más de 60 líneas. Para determinar este punto no se considerarán métodos para construir objetos, como por ejemplo un `switch` para construir las habilidades o métodos relacionados.

Hacer solo una cosa

Descuento máximo por tema: `ilimitado`

> Las funciones deberían hacer solo una cosa, y hacerla bien. Esto implica que las funciones debiesen ser pequeñas. Sólo se debe descontar en caso de que una función haga más de una cosa.

Descuentos sugeridos por error:

- `-0.05` Menos de 15 líneas
- `-0.1` Desde 15 a 30 líneas.
- `-0.25` Desde 30 a 60 líneas.
- `-0.5` Desde 60 a 100 líneas.
- `-1.0` 100 líneas o más.

> Recuerden que utilizamos el número de líneas como un proxy a que el método está haciendo más de una sola cosa, por lo tanto, puede que hayan métodos de

Código Duplicado

Descuento máximo por tema: `ilimitado`

> No debiese haber código duplicado entre funciones.

Descuentos sugeridos por error:

`-0.1` Por cada bloque de código duplicado.

Funciones booleanas

Descuentos máximos por tema: `0.5`

`-0.05` Por cada método que reciba un `bool`.

Condiciones en `if` y `while`

> más de 15 líneas que estén haciendo una sola cosa.

Argumentos

Descuento máximo por tema: `0.5`

> Las funciones deben recibir la menor cantidad de argumentos posibles, ojalá 1, máximo 3, en ningún caso más de 3.

Descuentos sugeridos por error:

`-0.05` Por función con más de 3 argumentos.

`-0.05` Por función que utilice out arguments.

Descuento máximo por tema: `0.5`

> Los booleanos encontrados en ifs, whiles u otros deben esta encapsulados en su propia función.

Descuentos sugeridos por error:

`-0.05` Por condición no encapsulada en una función.

Indentación

Descuentos máximos por tema: `0.5`

> Se puede tener como máximo **2** niveles de indentación en los métodos y funciones.

Descuentos sugeridos por error:

`-0.05` Por cada método con más de 2 niveles de indentación.

Cap 6

Condiciones Punto Base

- El diseño de las habilidad o de las afinidades respeta el Open-Close Principle.

Polimorfismo

Descuento máximo por tema: `ilimitado`

Métodos/atributos públicos y privados

> Es importante que en ciertos casos se modelen partes del juego usando polimorfismo.

Descuentos sugeridos por error:

`-0.8` Por utilizar programación funcional en las afinidades.

`-0.8` Por utilizar programación funcional en los efectos.

`-0.4` Por utilizar programación funcional en la selección de targets.

`-0.5` Por tener una modelación que hace que explote la cantidad de clases o que haya código duplicado entre clases. Como por ejemplo, tener una clase por habilidad.

`-0.1` Si usa polimorfismo pero rompe el Principio de Sustitución de Liskov.

`-0.1` No declarar métodos o atributos en clases abstractas o clases padres que deberían ser declarados, es decir, que todos sus hijos las declaran.

`-0.1` Por definir todos los métodos de una clase como virtuales aún cuando no todos son sobrescritos.

`-0.1` Métodos sin implementación que no son declarados como abstractos en clase padre/abstracta.

`-0.1` Clases que deberían ser abstractas que no son declaradas como abstractas. Por ejemplo, una clase concreta de la cual solo se usan sus subtipos.

Híbridos

> Es importante que se haga una distinción entre que debe ser público y que debe ser privado en un objeto.

Descuento máximo por tema: `0.5`

Descuentos por error:

`-0.05` Por método con un access modifier incorrecto.

`-0.01` Por atributo público en un objeto, el cual es utilizado sólo por la misma clase o sub clases (notar que si el atributo es utilizado por una clase externa, esta clase pasa a ser un híbrido y, por lo tanto, debería aplicarse ese descuento).

Train Wrecks

Descuento máximo por tema: `0.5`

> Un módulo no debería conocer los interiores del objeto que manipula.

Descuentos por error:

`-0.05` Por cada train wreck que no rompa la ley de Demeter, que tenga **más de 2 puntos** (`unit.Stats.HP` no se penaliza).

Ley de Demeter

Descuento máximo por tema: 1.0

> Una clase debe representar un objeto (esconde su estructura interna y expone abstracciones) o una EDD (expone su estructura interna, pero no implementa funcionalidad importante).

Descuentos por error:

-0.2 Por cada clase híbrida.

Descuento máximo por tema: 0.5

> Desde un método f dentro de una clase C solo se pueden llamar a métodos definidos en C y a métodos definidos en objetos que son creados en f, dados como argumentos a f o que son atributos de C. No se deben llamar a métodos definidos en objetos que son retornados en f.

Descuentos por error:

-0.1 Por cada violación a la ley.

Importante:

- No se deben considerar los casos en los que se llaman métodos de estructura de datos en vez de objetos. Estos no rompen Demeter.

Cap 10

Responsabilidades de la clase

Descuento máximo por tema: ilimitado

Descuentos sugeridos por error:

-0.1 Por cada responsabilidad extra dentro de una clase.

MVC

Separación de clases

> Se debe hacer una separación clara de las clases pertenecientes a cada tipo de proyecto, ya sea en carpetas o en proyectos distintos.

Descuentos máximos por error:

- `-0.5` No se distinguen qué clases pertenecen a cada tipo de proyecto.

Dependencias incorrectas

En cada uno de los casos siguientes hay un descuento máximo de `-0.5`.

- `-0.05` Por casos en que el proyecto modelo es dependiente del proyecto vista.
- `-0.05` Por casos en que el proyecto modelo es dependiente del proyecto controlador.
- `-0.05` Por casos en que el proyecto vista es dependiente del proyecto controlador.
- `-0.05` Por casos en que, dentro del proyecto vista, se utilizan métodos del modelo que editan los datos.
- `-0.05` Por casos en los que se tiene lógica específica de tener una vista por consola en el controlador.