Orientación a Objetos II 2025

Explicación de práctica
Semana del 21 de abril

Ejercicios de la semana

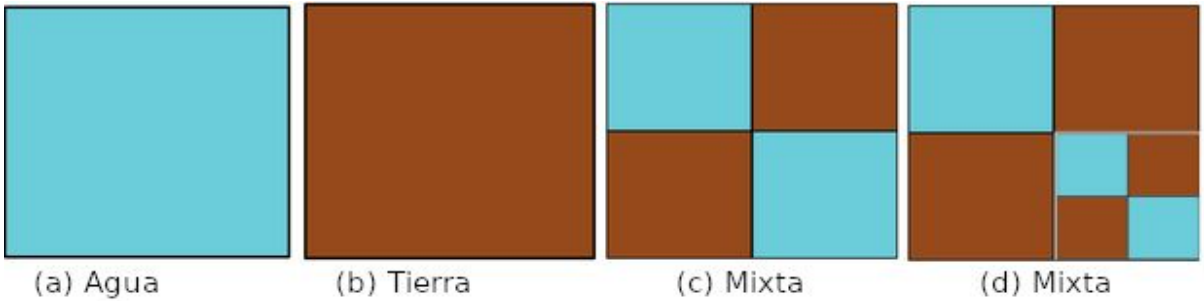
- Ejercicio 10: Mensajero
- Ejercicio 11: Topografias
- Ejercicio 12: FileSystem
- Ejercicio 13: SubteWay
- Ejercicio 14: Prestamos prendarios
- Ejercicio 15: Armado de PCs

Ejercicio 11 -Topografías

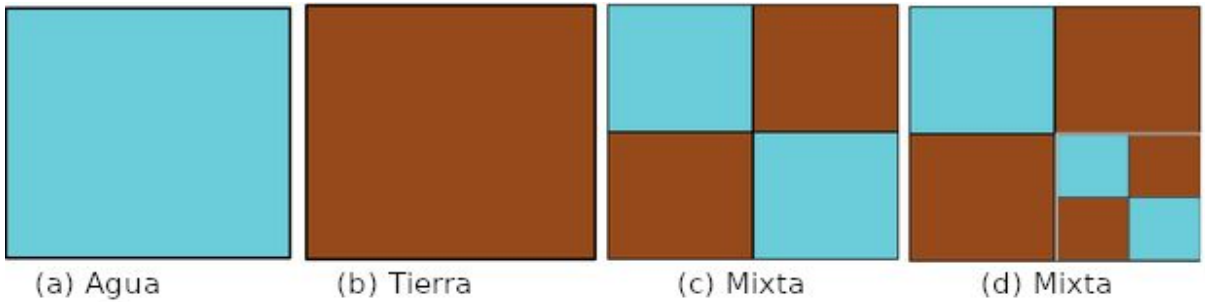
Un uso común de imágenes satelitales es el estudio de las cuencas hídricas que incluye saber la proporción entre la parte seca y la parte bajo agua. En general las imágenes satelitales están divididas en celdas. Las celdas son imágenes digitales (con píxeles) de las cuales se quiere extraer su “topología”.
Un objeto Topografía representa la distribución de agua y tierra de una celda satelital, la cual está formada por porciones de “agua” y de “tierra”. La siguiente figura muestra:

(a) el aspecto de una topografía formada únicamente por agua.
(b) otra formada solamente por tierra.
(c) y (d) topografías mixtas.

Ejercicio 11 -Topografías



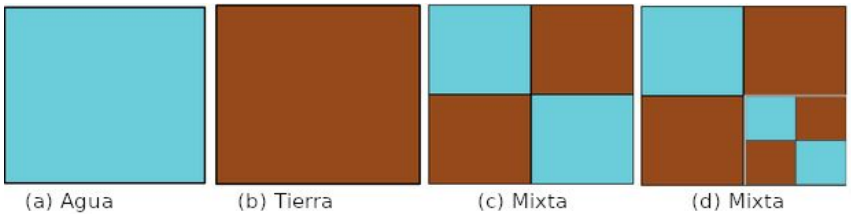
Ejercicio 11 -Topografías



(a) Topografía íntegramente de Agua
(b) Topografía íntegramente de Tierra
(c) Topografía Mixta formada por
  (i) Topografía íntegramente de Agua
  (ii) Topografía íntegramente de Tierra
  (iii) Topografía íntegramente de Tierra
  (iv) Topografía íntegramente de Agua

(d) Topografía Mixta formada por
  (i) Topografía íntegramente de Agua
  (ii) Topografía íntegramente de Tierra
  (iii) Topografía íntegramente de Tierra
  (iv) Topografía Mixta formada por
    (1) Topografía íntegramente de Agua
    (2) Topografía íntegramente de Tierra
    (3) Topografía íntegramente de Tierra
    (4) Topografía íntegramente de Agua

Ejercicio 11 - Proporción de Agua



(a) Topografía de Agua = 1
(b) Topografía de Tierra = 0
(c) Topografía Mixta
  (i) Topografía de Agua = 1
  (ii) Topografía de Tierra = 0
  (iii) Topografía de Tierra = 0
  (iv) Topografía de Agua = 1

= 2 / 4

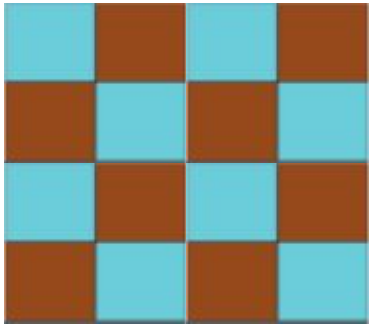
Topografía Mixta = 1/2

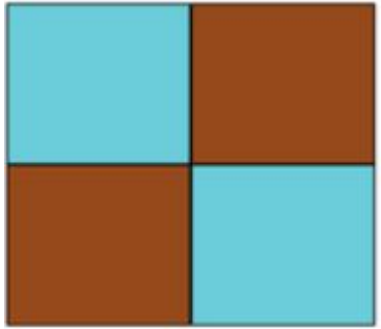
(d) Topografía Mixta
  (i) Topografía de Agua = 1
  (ii) Topografía de Tierra = 0
  (iii) Topografía de Tierra = 0
  (iv) Topografía Mixta
    (1) Topografía de Agua = 1
    (2) Topografía de Tierra = 0
    (3) Topografía de Tierra = 0
    (4) Topografía de Agua = 1

= (1 + 0 + 0 + 1) / 4
= 1/2

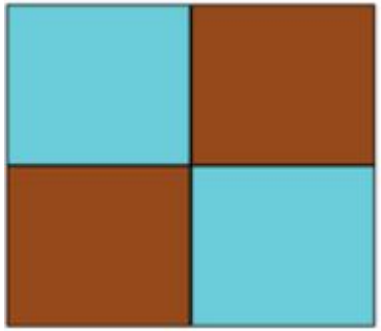
= (1 + 0 + 0 + 0.5) / 4
= 1.5 / 4

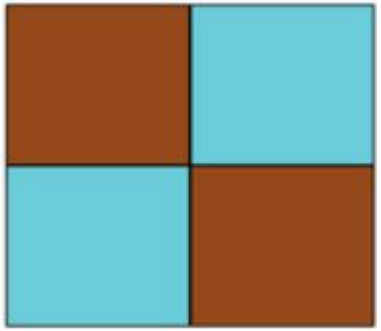
Ejercicio 11 - Igualdad





Misma proporción, muy diferentes





Misma proporción, distinto orden

Patrón Composite

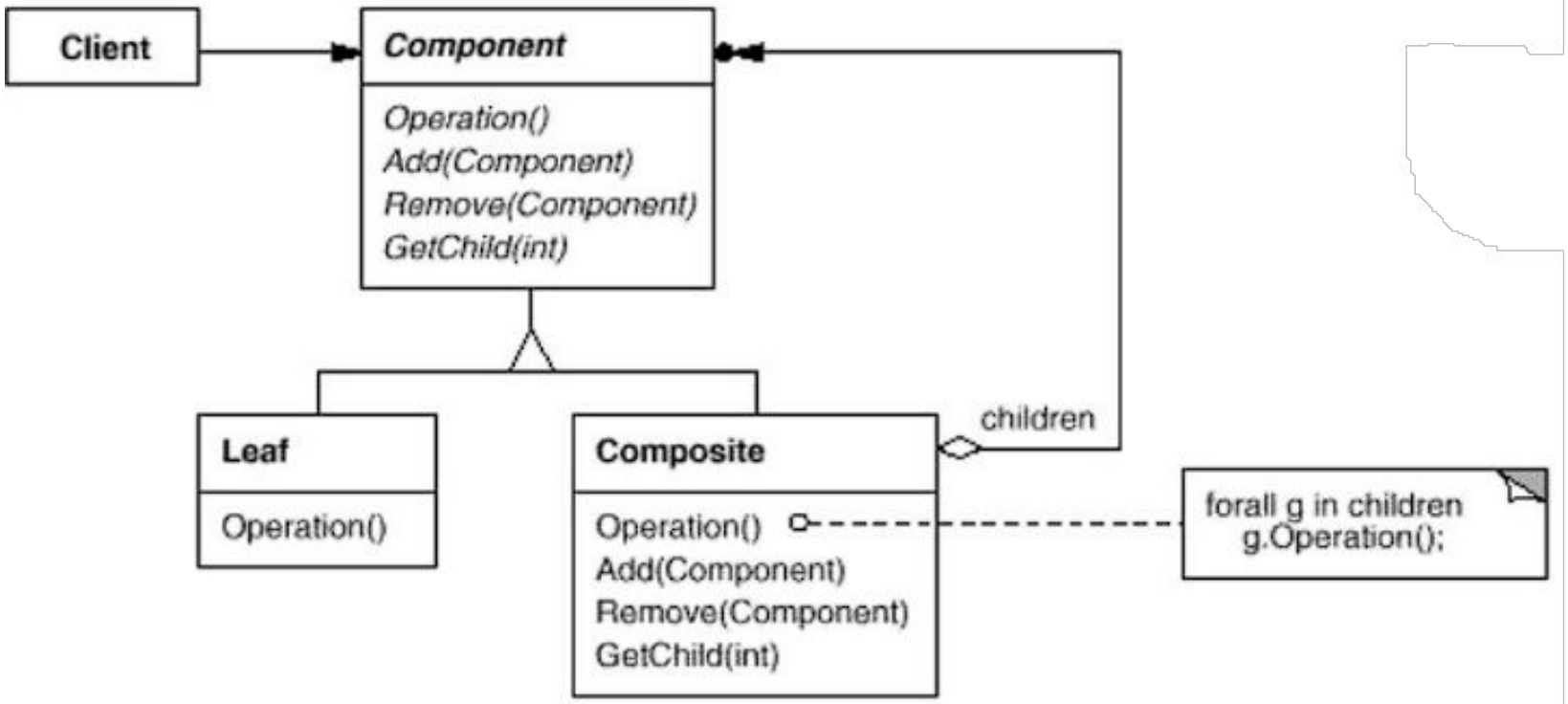
Propósito: Componer objetos en estructuras de árbol para representar jerarquías parte-todo.

El Composite permite que los clientes traten a los objetos atómicos y a sus composiciones uniformemente.

Aplicabilidad: Use el patrón Composite cuando:

- ✓ quiere representar jerarquías parte-todo de objetos.
- ✓ quiere que los objetos “clientes” puedan ignorar las diferencias entre composiciones y objetos individuales. Los clientes trataran a los objetos atómicos y compuestos uniformemente.

Patrón Composite - Estructura



Patrón Composite - Participantes

Component:

- Declara la interfaz para los objetos de la composición.
- Implementa comportamientos default para la interfaz común a todas las clases.
- Declara la interfaz para definir y acceder “partes de la composición”.

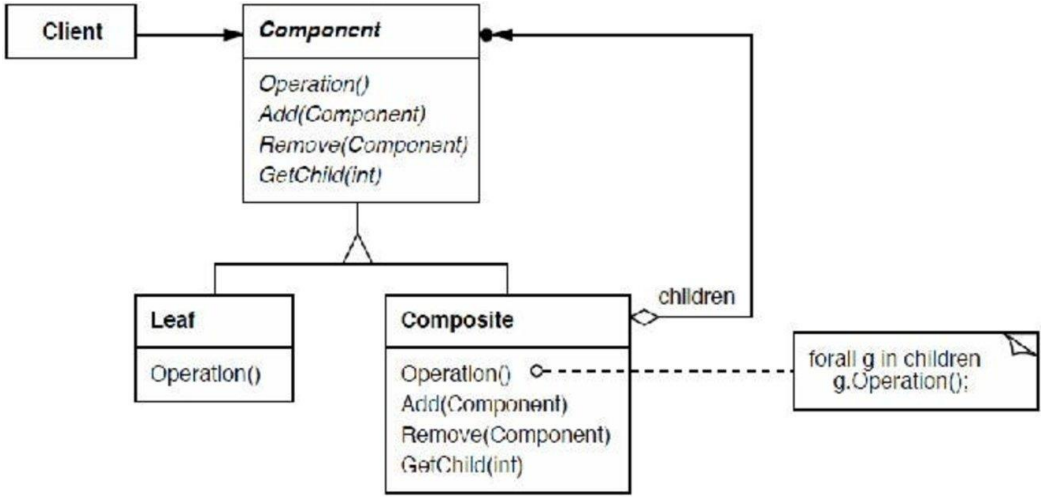
Leaf:

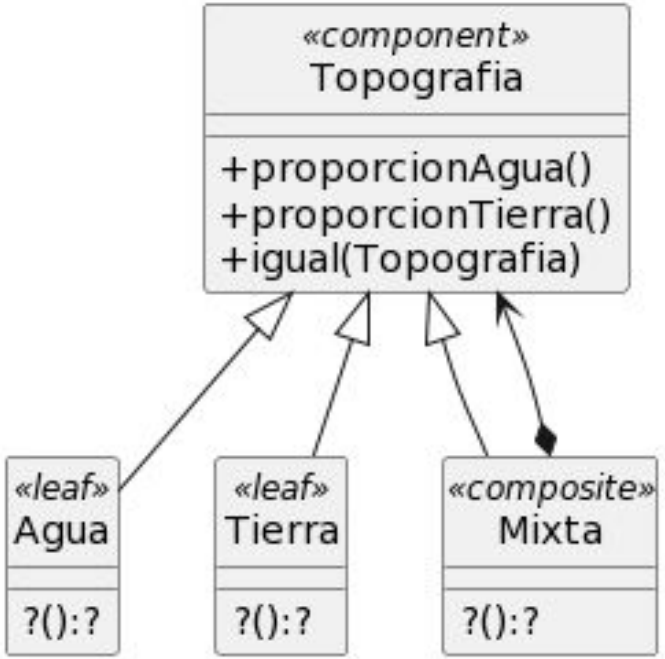
- Las hojas no tienen sub-árboles.
- Define el comportamiento de objetos primitivos en la composición.

Composite:

- Define el comportamiento para componentes complejos.
- Implementa operaciones para manejar el sub-árboles.

Ejercicio 11 - Diseño





Patrón Composite - Consecuencias

+ Define “jerarquías” de objetos primitivos y compuestos.
+ Los objetos primitivos pueden componerse en objetos complejos, los que a su vez pueden componerse recursivamente.
+ Simplifica los objetos cliente. Los clientes usualmente no saben (y no deberían preocuparse) acerca de si están manejando un compuesto o un simple.
+ Hace más fácil el agregado de nuevos tipos de componentes porque los clientes no tienen que cambiar cuando aparecen nuevas clases componentes.
- Debe ser “customizado” con reglas de composición (si fuera necesario)

Patrón Composite - Cuestiones de implementación

- Referencias explícitas a la raíz de una hoja
- Maximizar el protocolo de la clase/interfaz Component
- Orden de las hojas
- Borrado de componentes
- Búsqueda de componentes(fetch) por criterios
- Diferentes estructuras de datos para guardar componentes

Patrón Composite - Malos olores

- La jerarquía no es polimórfica
- Composite no tiene relación con otras clases de la jerarquía
- Composite no implementa add/remove
- Composite no delega en la colección

Ejercicio 13 - SubteWay

Una cadena de comidas rápidas especializada en sándwiches necesita resolver el cálculo de precios de éstos. El cálculo es simple: el precio de un sándwich equivale a la suma del precio de cada uno de sus componentes; el problema es la dificultad para representar y crear cada uno de los sándwiches distintos.

Existen cuatro sandwiches distintos, pero podrían aparecer nuevos en el futuro.

Ejercicio 13 - SubteWay

Clásico: consta de pan brioche (100 pesos), aderezo a base de mayonesa (20 pesos), principal de carne de ternera (300 pesos) y adicional de tomate (80 pesos).

Vegetariano: consta de pan con semillas (120 pesos), sin aderezo, principal de provoleta grillada (200 pesos) y adicional de berenjenas al escabeche (100 pesos).

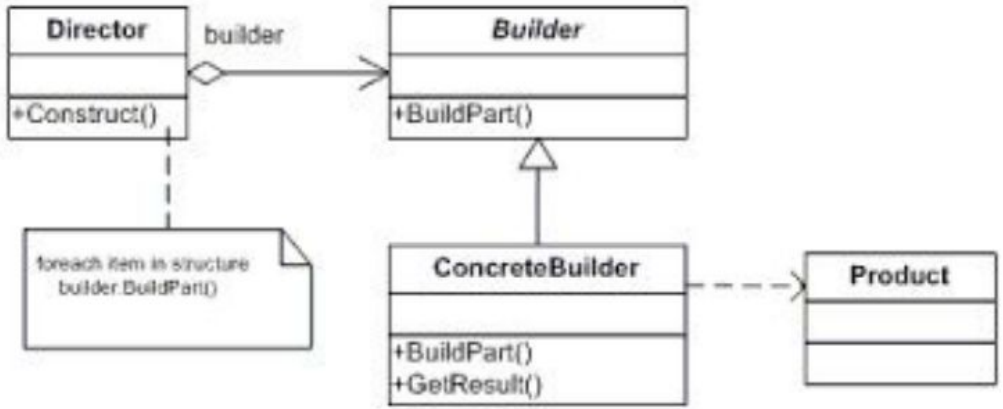
Vegano: consta de pan integral (100 pesos), aderezo de salsa criolla (20 pesos), principal de milanesa de girgolas (500 pesos), sin adicional.

Tareas:
Proponga un diseño e implementación para la creación y cálculo de precios de estas alternativas de sándwiches.
Si en su solución aplicó algún(os) patrón(es) de diseño, indique cuál(es) y sus roles.

Ejercicio 13 - SubteWay

Builder

- Intención: separa la construcción de un objeto complejo de su representación (implementación) de tal manera que el mismo proceso puede construir diferente representaciones (implementaciones)
- Estructura (**Roles**)



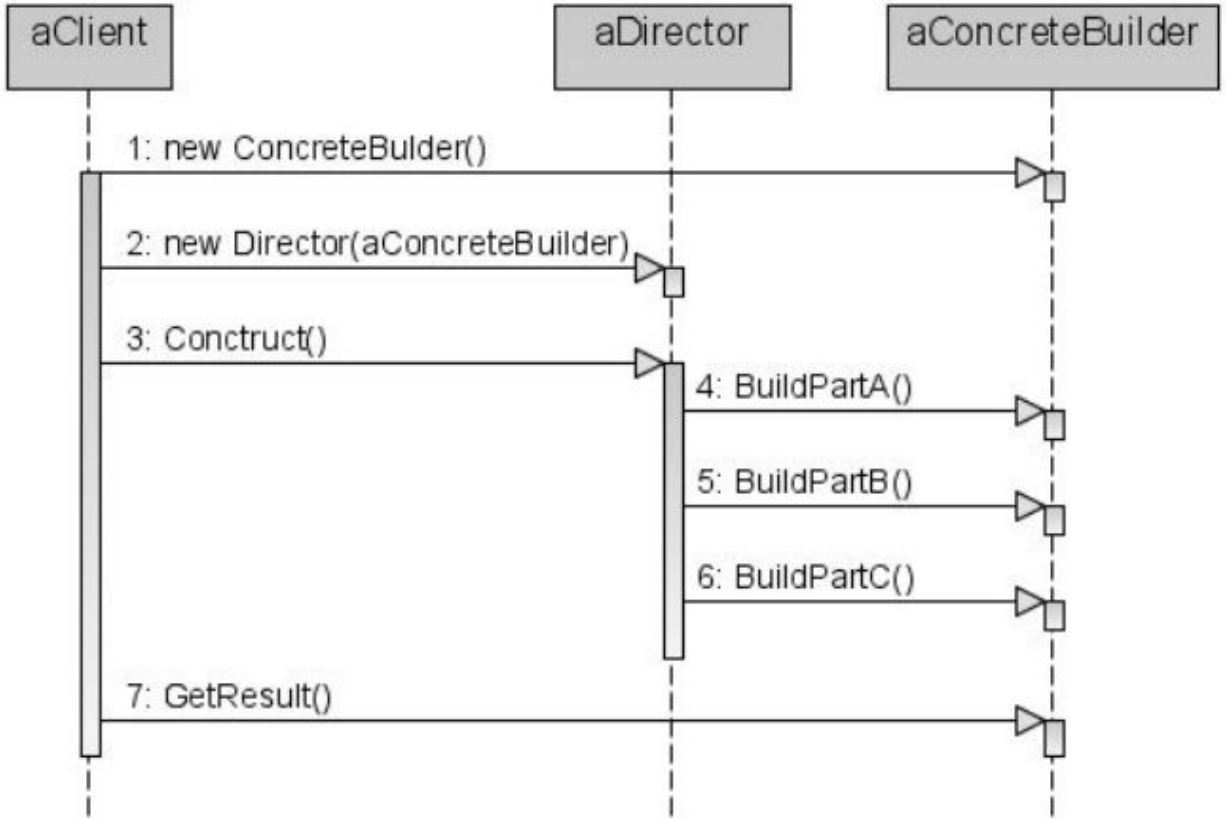
Ejercicio 13 - SubteWay

Builder

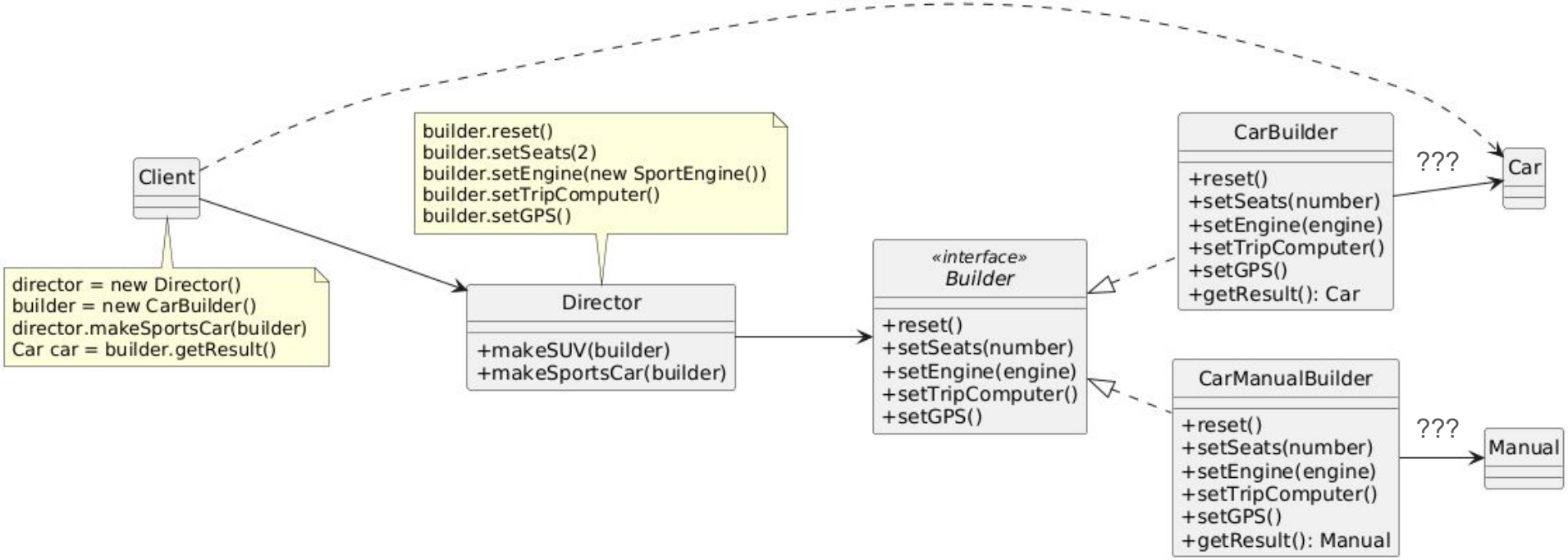
Participantes

- Builder: especifica una interface abstracta para crear partes de un Producto
- Concrete Builder: construye y ensambla partes del producto.
  - Guarda referencia al producto en construccion
- Director: conoce los pasos para construir el objeto
  - Utiliza el Builder para construir las partes que va ensamblando
  - En lugar de pasos fijos puede seguir una “especificación” (ver knwon uses@ GOF)
- Product: es el objeto complejo a ser construido

Ejercicio 13 - SubteWay



Ejercicio 13 - SubteWay



Ejercicio 13 - Builder

Vale la pena Builder?

Si!

- Abstrae la construcción compleja de un objeto complejo
- Permite variar lo que se construye Director <-> Builder
- Da control sobre los pasos de contrucción

No!

- Requiere diseñar y implementar varios roles
- Cada tipo de producto requiere un ConcreteBuilder
- Builder suelen cambiar o son parsers de specs (> complejidad)

Patrón Builder - Algunas consideraciones

- El director sólo sabe hacer una cosa
- Los builder pueden saber hacer cosas que no requiera un director, pero sí otro
- Diferentes directores pueden usar los mismos builders
- Cuando aparece una nueva receta => nuevo director
- Cuando aparecen nuevos productos => nuevo builder

Patrón Builder - Errores comunes

- Director construye partes
- Director necesita diferentes builder al mismo tiempo
- Builder manda mensajes al director
- Sólo hay un builder