

Patrones

Aplicables a Descripciones de Dominio

Patrón

Mejor solución a un problema recurrente



Lenguaje de Patrones: conjunto de patrones interrelacionados

Pattern Language (mini)

Problema: ante una urgencia un gran número de personas necesitan abandonar una sala rápidamente

Solución: instalar puertas que se abren para afuera de la sala

Problema: cuando la puerta se abre para afuera, existe la chance de golpear a los que pasan por el pasillo

Solución: ?





Helicopteros

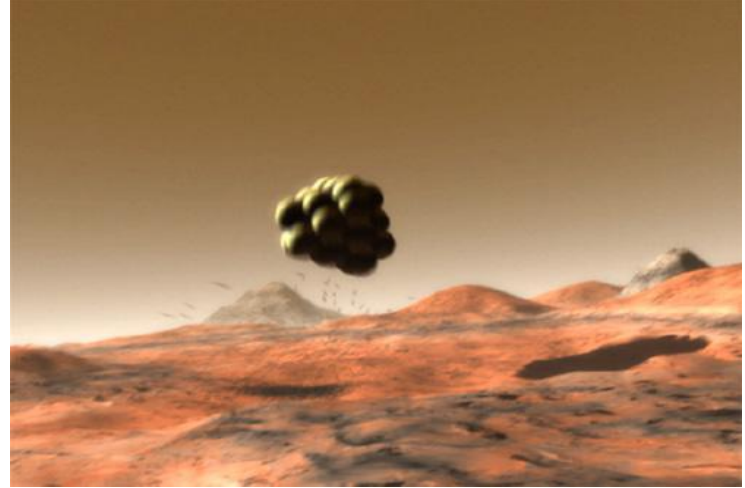
- **Problema:** la energía de impacto vertical suele dañar la columna de los tripulantes de helicópteros
- **Objetivo:** reducir la energía del impacto
- **Solución:** agregar contenedores de un gas inerte para absorber parte de la energía



Colateral: las bolsas desplegadas no son aerodinámicas

Viaje Espacial

- **Problema:** la energía de impacto dañaría el robot de exploración de Marte
- **Objetivo:** reducir la energía del impacto. Retro-rockets suelen fallar y son caros
- **Solución:** construir una coraza con pelotas infladas con un gas que absorba parte de la energía del impacto.



Colateral: las bolsas desplegadas no soportan el lanzamiento por cohete desde la tierra

Automotores

- **Problema:** la energía generada por un choque frontal suele producir que la cabeza del conductor impacte en la columna de dirección
- **Objetivo:** reducir la energía del impacto de la cabeza
- **Solución:** air-bag



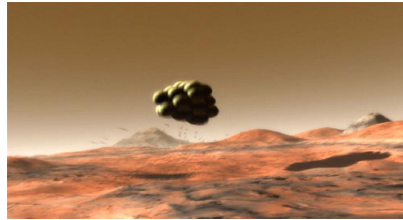
Colateral: el air-bag desplegado no permiten manejar

Patrón AirBag

Problema: El contenido de un vehículo sufre daños catastróficos ante un impacto severo.

Objetivo: reducir la energía del impacto en el vehículo o su contenido

Solución: Utilizar contenedores con una sustancia compresible (gas) para absorber la energía del impacto a niveles aceptables.



Consecuencia: el contenedor de gas solo es necesario durante el impacto (esto será otro patrón?)



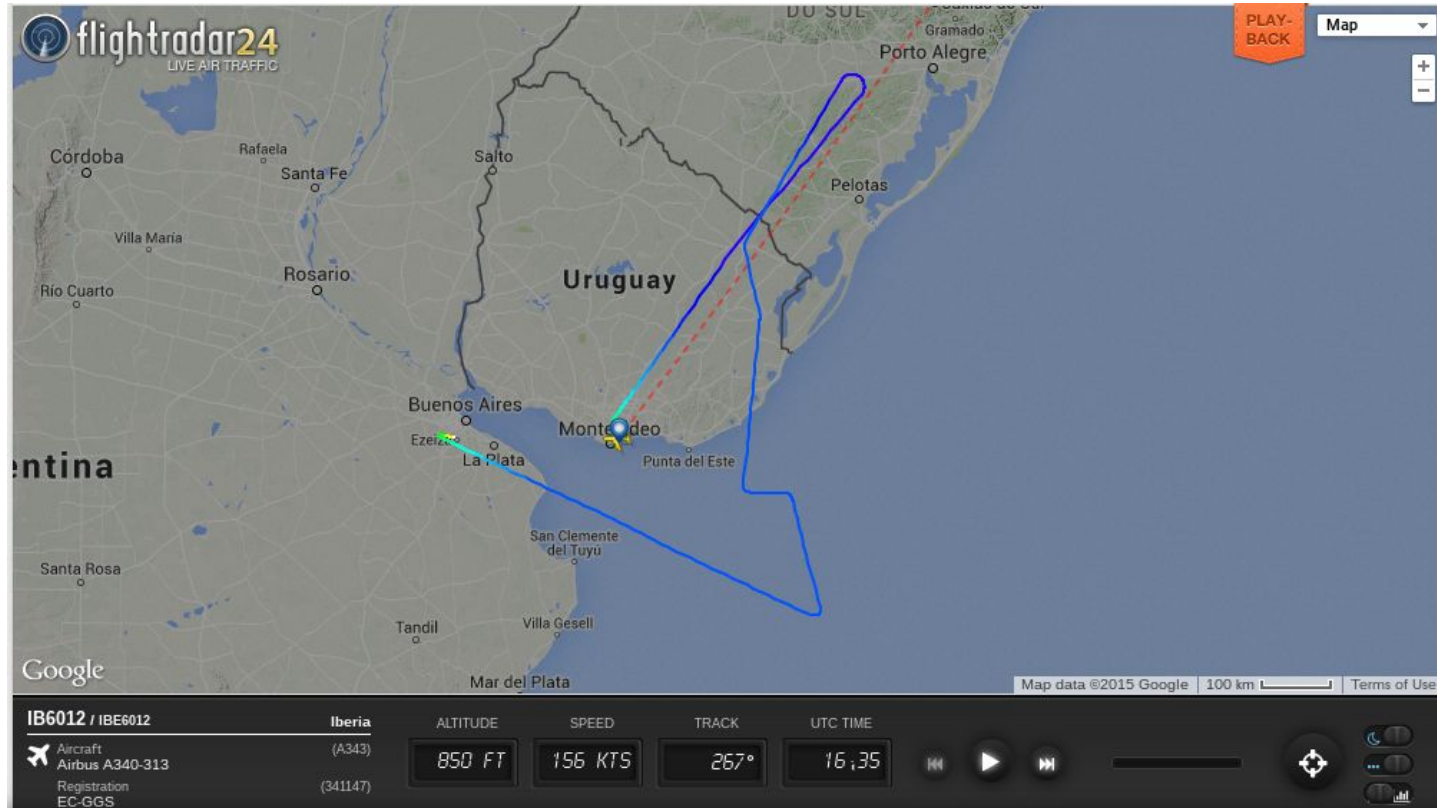
Dos
maneras
de hacer
una torta
sobre
Castillos



Engine Flameout

- Es un incidente que resulta en el malfuncionamiento de un motor jet
- El entrenamiento de un piloto incluye conocer la mejor solución a este tipo de problemas
- Diferentes aviones tienen diferentes procedimientos
- Diferentes momentos del vuelo requieren diferentes procedimientos
 - Carreteo
 - Despegue
 - Vuelo
- **Importante:**
 - Reconocer el problema de manera exacta
 - Encontrar la solución correcta
 - Aplicar la solución de manera apropiada

IB6012 (18/08/2015)





US Airway 1549



<https://youtu.be/JSIbHK07fkY?t=68>

Design Patterns: smells más comunes

Patron	Smell
Template Method	<ol style="list-style-type: none">1. No existe una definición única de algoritmo u operación2. Subclasses invocan "super msg"
Strategy	<ol style="list-style-type: none">1. Strategies subclase de la clase "cliente"2. Strategies no son polimorficas3. Strategies con estado4. Implementa más de un método público (responsabilidad / algoritmo)
State	<ol style="list-style-type: none">1. Estados subclases del "cliente"2. Se pregunta por nombre o tipo de estado3. Estados no polimorficos4. No hay cambio de transiciones de estado5. Los estados no implementan comportamiento diferente

Patron	Smell
Composite	<ol style="list-style-type: none">1. No relación con otras clases de Jerarquía2. No implementa add/remove3. Composite no es polimorfico4. Composite no delega en la coleccion
Façade	<ol style="list-style-type: none">1. "Cliente" interactúa con los componentes2. No hay coordinación de las partes
Adapter	<ol style="list-style-type: none">1. No respeta el protocolo esperado2. Subclase del "adaptado"
Command	<ol style="list-style-type: none">1. Los comandos no son polimórficos2. Existen inter-dependencias3. No existe el "procesador de comandos"
Builder	<ol style="list-style-type: none">1. El director construye partes2. Director necesita diferentes builders3. Confunde con FactoryMethod