

Status 428: Precondition Required

# Módulo 3: REST en la práctica

---

# Repaso hasta acá

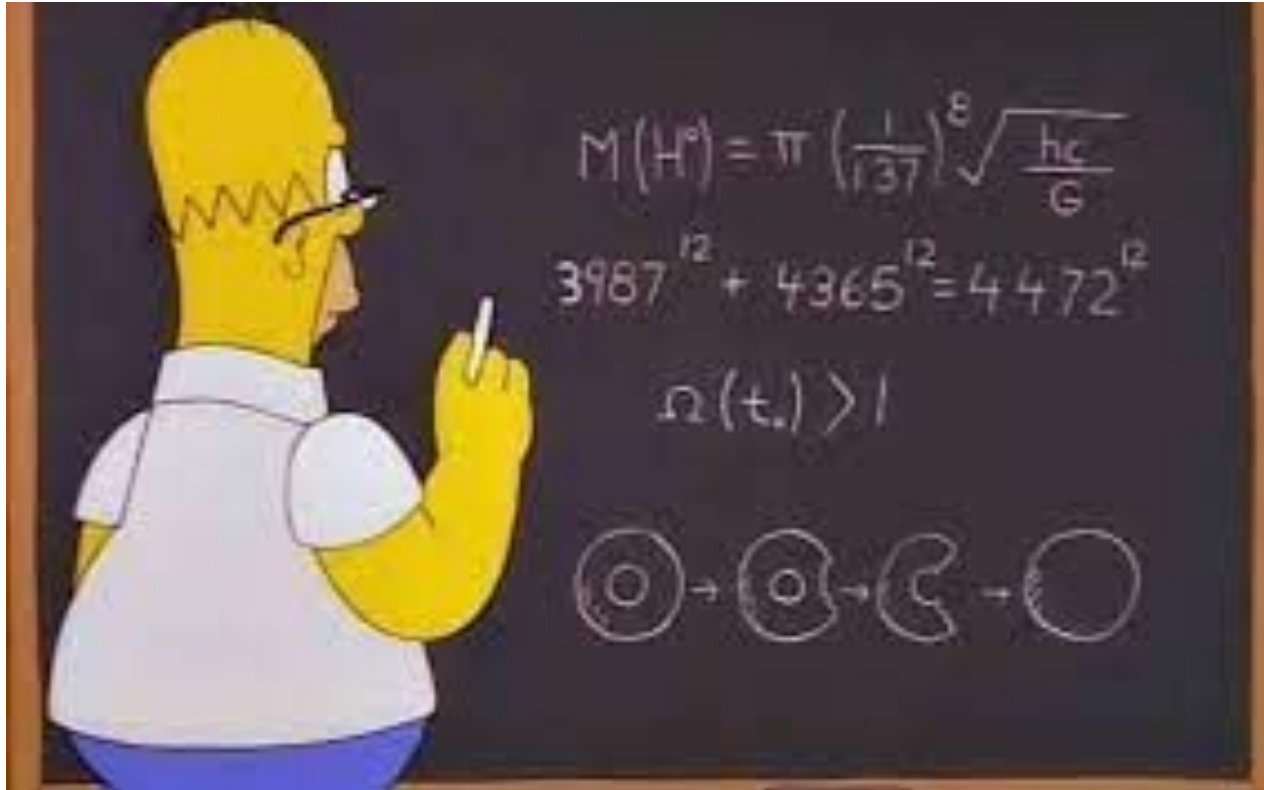
Evolución de las plataformas de cómputo:

- Software monolítico, sistemas operativos, bibliotecas/frameworks, IPC, RPC

Surge la Web

- Hypermedia para documentos
- Hypermedia como plataforma de aplicaciones: REST
  - Modelo de Richardson
  - Buenas prácticas: qué hacer y no hacer a la hora de diseñar APIs REST

Muy bonita toda la teoría...



¡Vayamos a tomar un helado!



# Heladería Via Apilia

## Requerimientos:

- Listar los gustos de helado
  - Con opción de filtrado según tipo de gusto (cremas, chocolates, ...)
- Poder realizar pedidos
  - Un pedido consta de una dirección de envío y múltiples pots
  - Cada pote tiene un tamaño (250gr, 500gr, 1kg) y lleva varios gustos
  - Podemos **agregar/quitar pots de un pedido a medida que se lo arma**
  - Podemos **cambiar** la dirección de envío
  - Podemos ver los detalles de un pedido
- Poder pagar pedidos de manera asincrónica
  - Se proveen los datos de pago y se puede consultar el estado de la transacción

# Code first vs API first

## Code first

Ante estos requerimientos, empezamos a implementar nuestra solución

## API first



Empezamos por definir un contrato para nuestra API y **luego** implementaremos

# Desafíos

- Payloads en operación **casi** simétricas como por ejemplo POST pedido y su respuesta.
- Cómo modelar operaciones que no se llevan con ningún verbo REST como “pagar”
  - Sobre todo cuando son operaciones asincrónica
  - Patrón POST/GET/GET
- Cómo hacer para llegar al nivel 3 de Richardson (HATEOAS)
  - La dificultad de que no hay un **buen** standard de hypermedia links
  - Cómo hacer para modelar el verbo, parámetros, etc.
- La clave del reuso al modelar: reuso de parámetros, de schemas, etc.