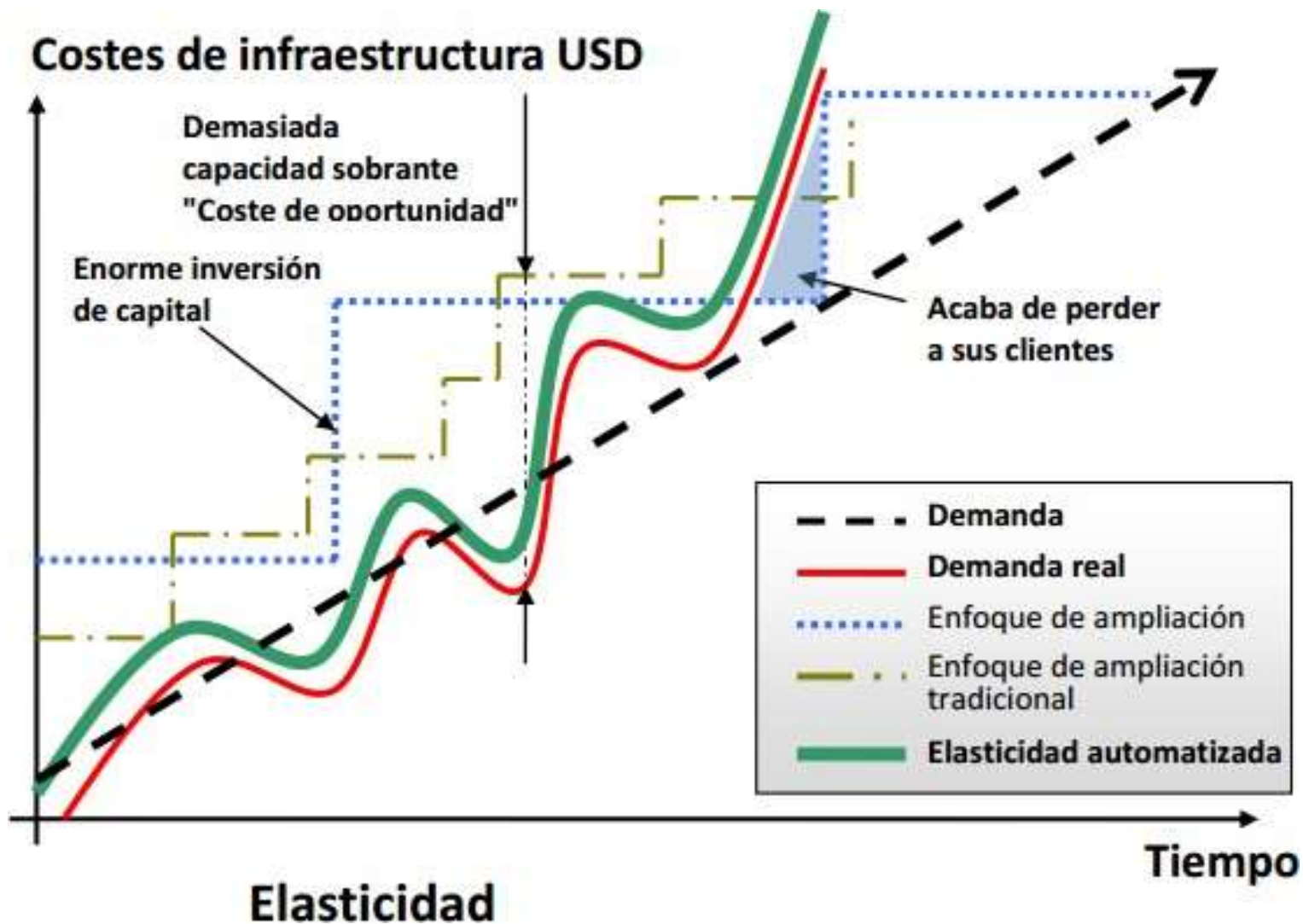
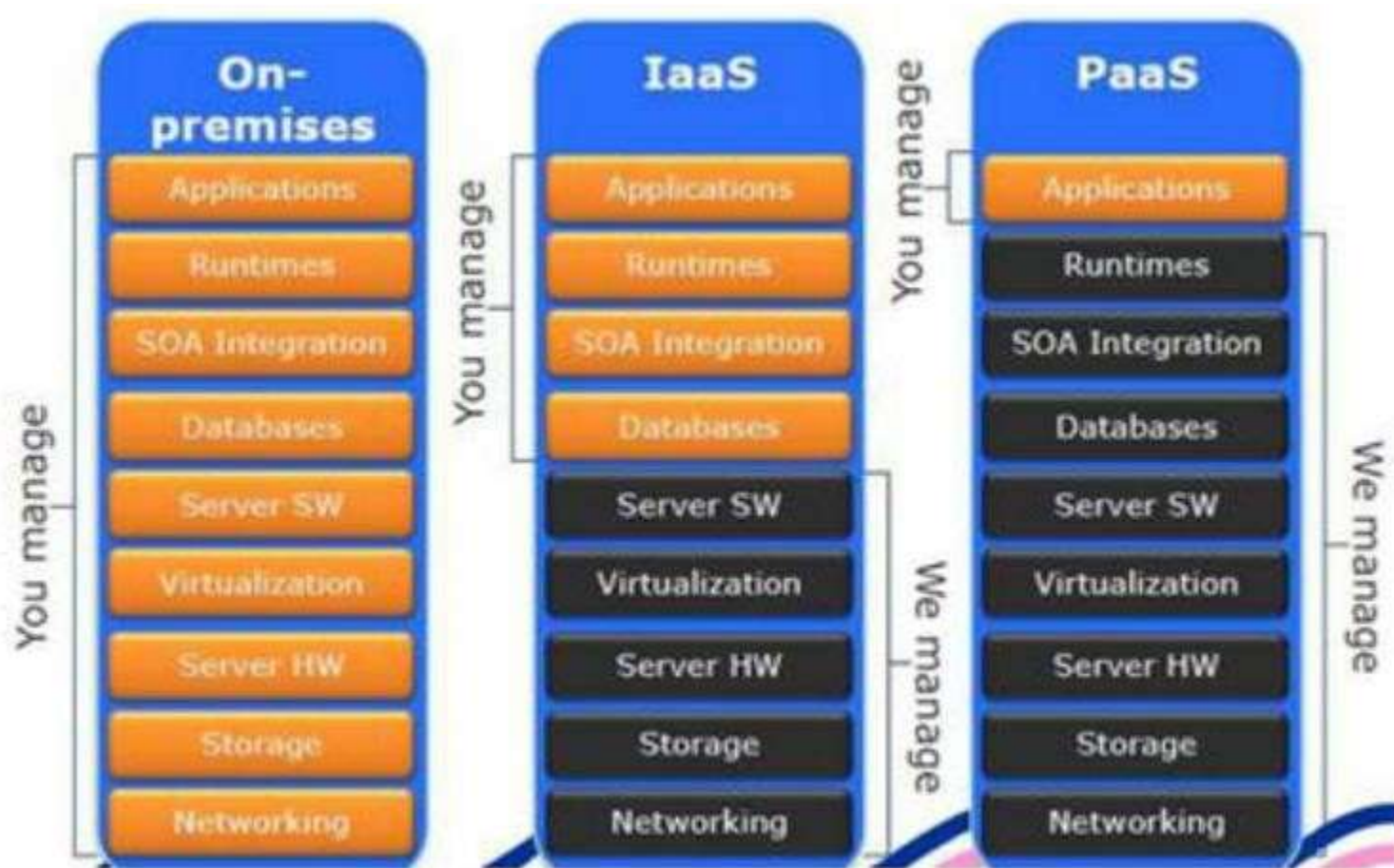


# EL ESTILO CLOUD

---



# ¿ Qué es Cloud Computing ?



# Cloud Computing

"Anything that can go wrong, will go wrong." - Murphy's Law

- ① Design for failure (resilience)
- ② Design for availability
- ③ Design for elasticity
- ④ Design for the future

# Escalabilidad

Escalabilidad Vertical → Scale UP ← Implica un **DownTime** ... ¿ por qué ?

Escalabilidad Horizontal → Scale OUT ← Es más complicada ... ¿ por qué ?

- Generalmente los «nodos» cumplen funciones específicas
- Es más eficiente cuando los nodos son «homogéneos»

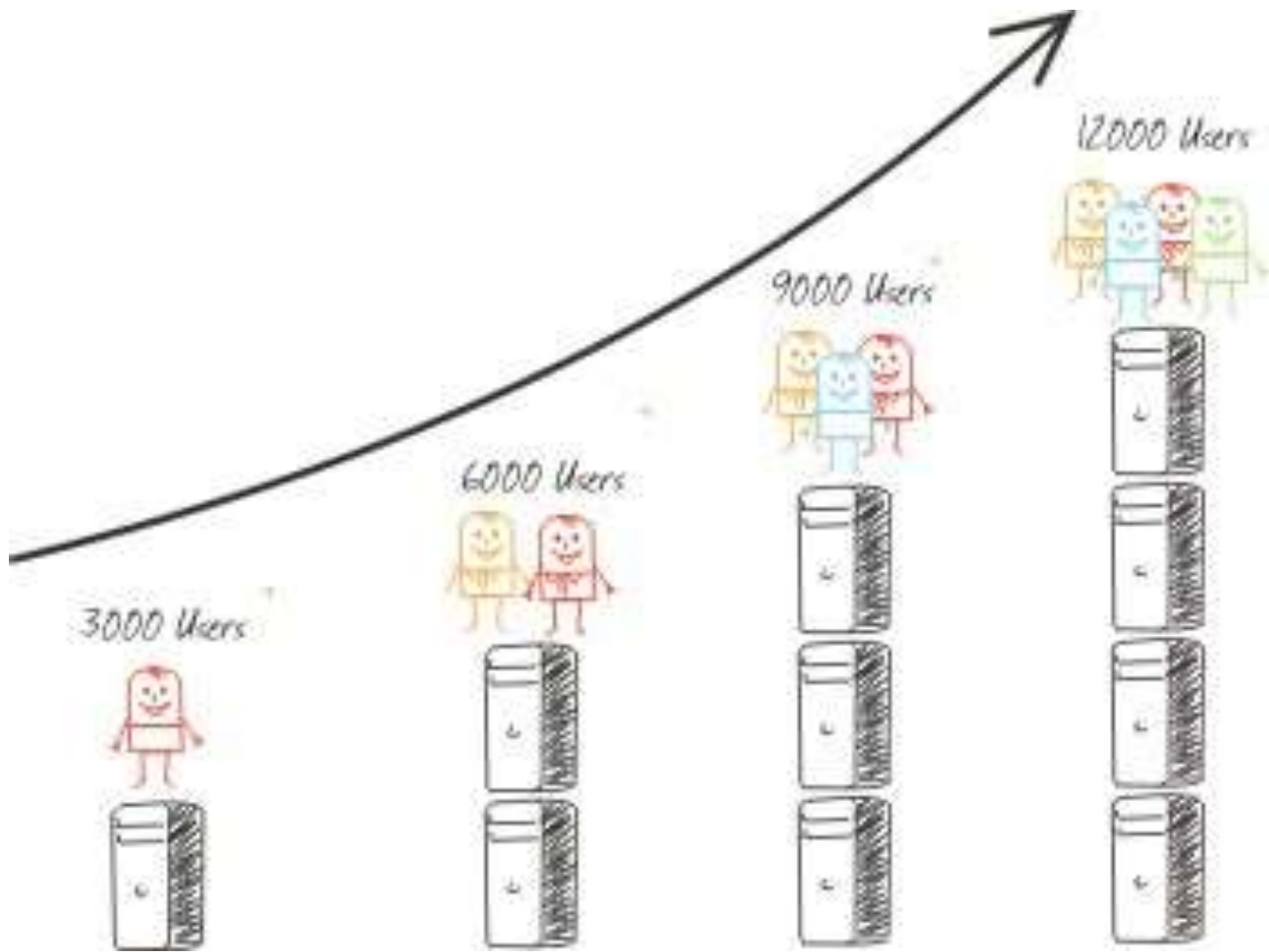
Describir la escalabilidad: Con 100 usuarios concurrentes, el TR debe estar debajo de 2 segundos el 60% del tiempo, de 2 a 5 segundos 38% del tiempo y 5 segundos o más el 2% del tiempo.

Describir el «workload»

Describir la «unidad de escalamiento» : ejemplo: por cada 100 usuarios se requieren 2 nodos

Límite de la escalabilidad : cuellos de botella en los «recursos»

- Mejora por algoritmo ← Ejemplos ?
- Mejor por Hardware



# Performance

**Performance** → es la experiencia de un usuario individual → se asocia al tiempo de respuesta

**Escalabilidad** → es la experiencia de un grupo de usuarios

# Patrón: Horizontally Scaling Compute

## Contexto:

- Ahorro de costos
- Los requerimientos de capacidad de la aplicación exceden la capacidad actual de los nodos
- Los requerimientos de capacidad de la aplicación varían estacionalmente o están sujetos a picos impredecibles.
- Los nodos de cómputo requieren un mínimo de downtime o resiliencia en caso de fallas de HW, actualizaciones de sistema o cambios en recursos.

## Impacto:

- Disponibilidad
- Optimización de costos
- Escalabilidad
- Experiencia de Usuario

Concepto importante : Manejo de Sesión



# Patrón : Queue-Centric Workflow

Subconjunto de CQRS (Command and Query Responsibility Segregation)

Sólo aplica para Updates ( aplicaciones web y/o móviles)

Contexto:

- La aplicación está desacoplada en diversos «tiers», pero los «tiers» necesitan colaborar.
- La aplicación necesita garantizar por lo menos un procesamiento de mensajes a lo largo de los «tiers»
- Se espera una experiencia de usuario «responsive» en el «tier» de GUI aun y cuando haya servicios propios o de terceros que se usen durante el procesamiento.

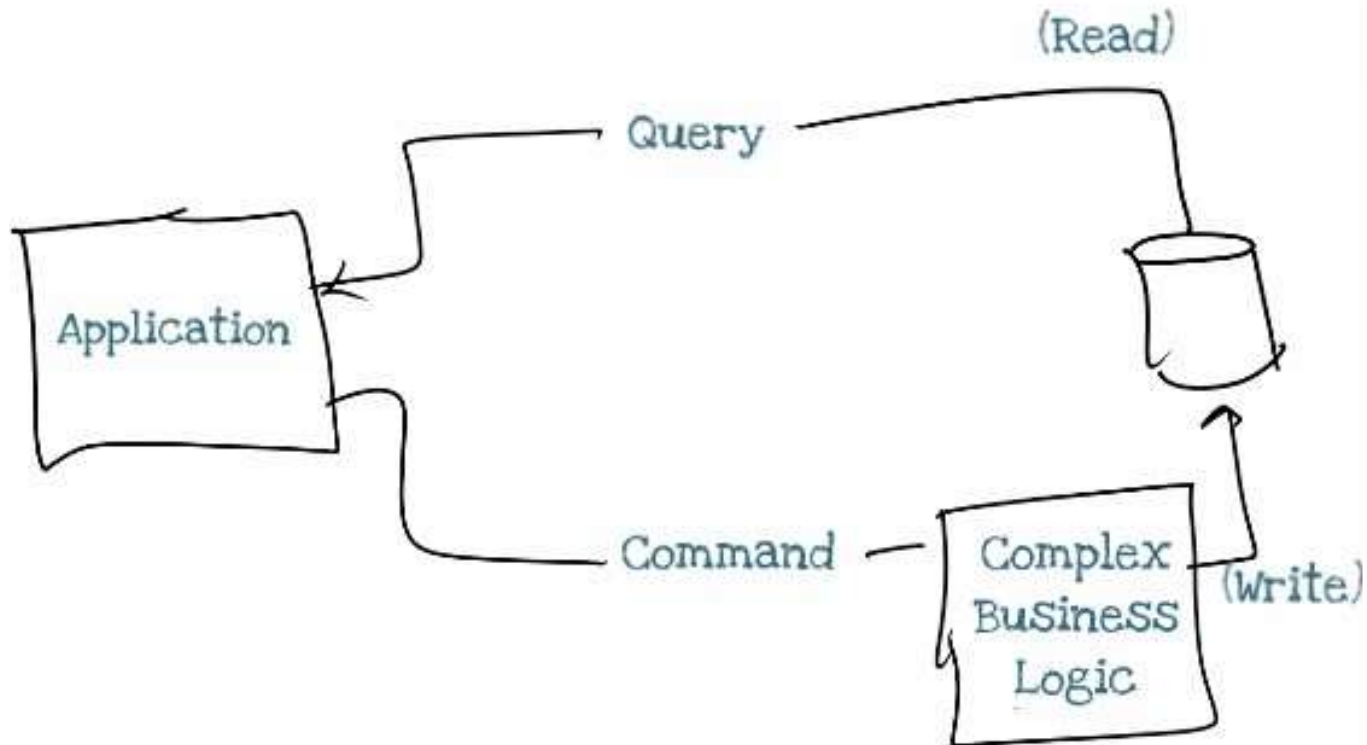
Impacto:

- Disponibilidad
- Optimización de costos
- Escalabilidad
- Experiencia de Usuario

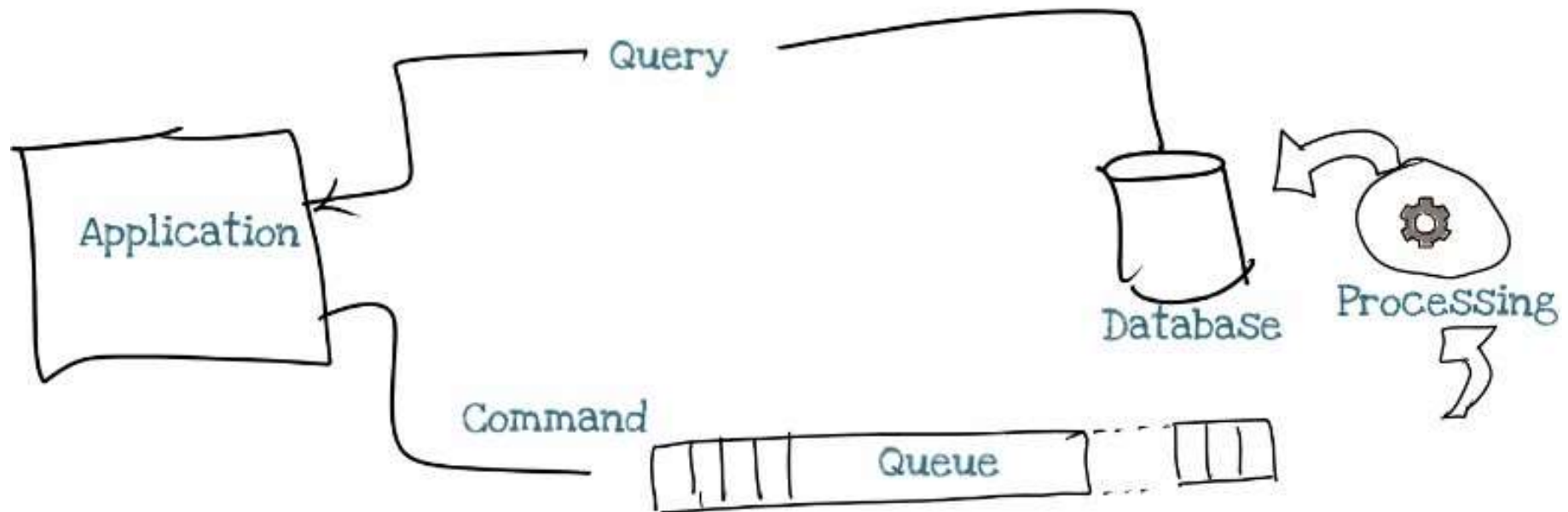
Conceptos importantes :

- Comunicación Asíncrona
- Bajo acoplamiento

## Patrón CQRS



## Patrón Queue-Centric

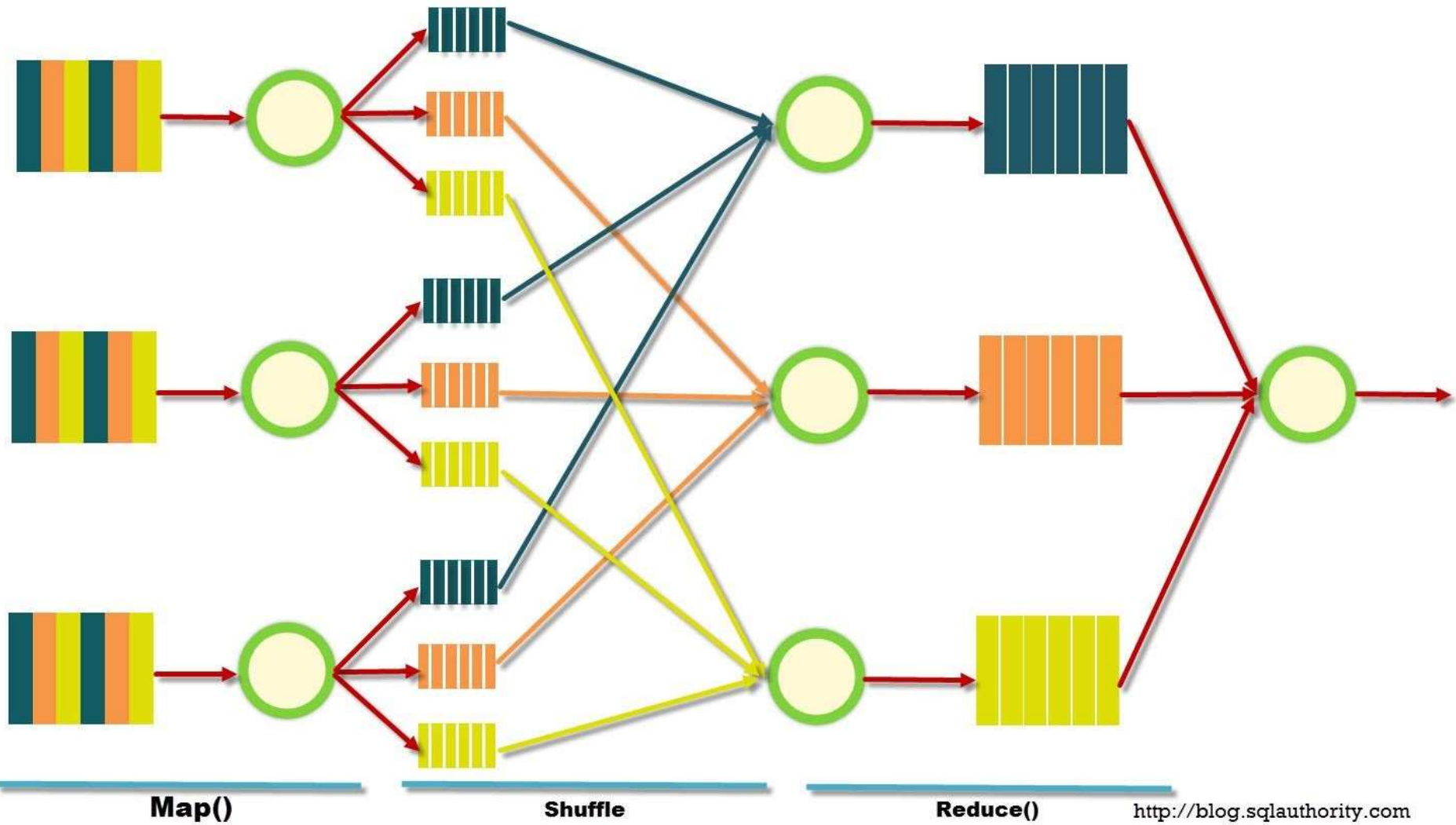


# Patrón: Map-Reduce

## Contexto:

- La aplicación procesa grandes volúmenes de datos estructurados
- La aplicación procesa grandes volúmenes de datos semi-estructurados
- Los requerimientos de análisis de los datos cambian frecuentemente o son ad-hoc
- La aplicación requiere generar reportes que los sistemas tradicionales no pueden procesar eficientemente
- Requiere tener dos funciones : mapper y reducer
- Esta diseñado para procesamiento batch

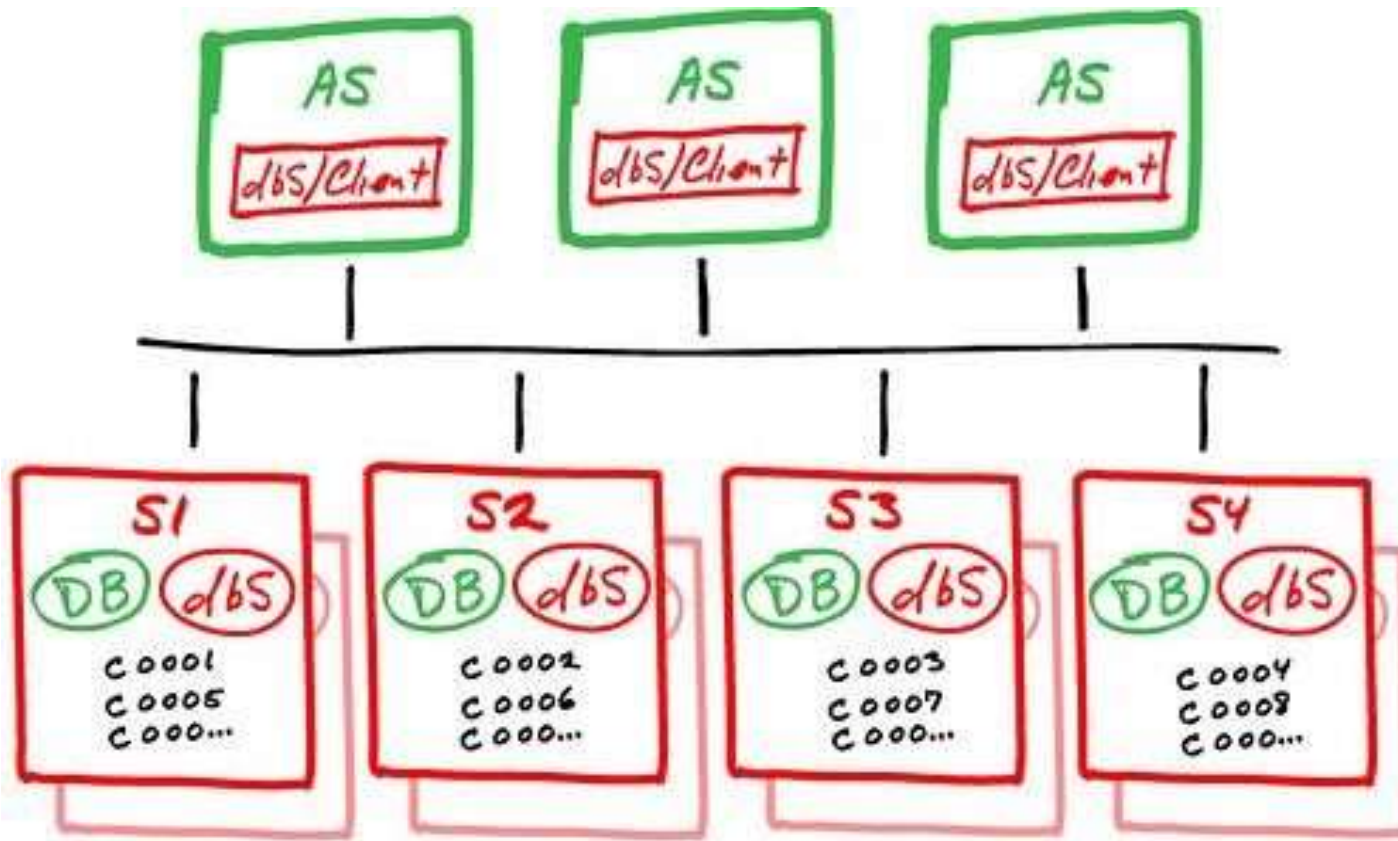
## How MapReduce Works?



# Patrón: Database Sharding

## Contexto:

- El volumen de consultas de una aplicación excede la capacidad de un nodo de base de datos resultando en un tiempo de respuesta elevado.
- El volumen de transacciones de una aplicación excede la capacidad de un nodo de base de datos resultando en un tiempo de respuesta elevado.
- El consumo de ancho de banda de una aplicación excede el ancho de banda disponible de un nodo de base de datos resultando en un tiempo de respuesta elevado.
- Los requerimientos de almacenamiento exceden la capacidad de un nodo de base de datos.
- Este patrón es una aproximación de escalamiento horizontal que sobrepasa el límite de capacidad de un nodo de base de datos al distribuir la base de datos a lo largo de múltiples nodos.
- Cada nodo contiene un subconjunto de los datos ( shard ).
- La data de todos los “shards” representa una base de datos completa.



- Identificación de shard: una columna específica se designa como la “shard key” y determina en que nodo se almacena el registro.
- La “shard key” es necesaria para acceder a los datos.
- No todas las tablas van al sharding

# 12 Factor App

<https://12factor.net/es/>

## **I. Código base (Codebase)**

Un código base sobre el que hacer el control de versiones y múltiples despliegues

## **II. Dependencias**

Declarar y aislar explícitamente las dependencias

## **III. Configuraciones**

Guardar la configuración en el entorno

## **IV. Backing services**

Tratar a los “backing services” como recursos conectables

## **V. Construir, desplegar, ejecutar**

Separar completamente la etapa de construcción de la etapa de ejecución

## **VI. Procesos**

Ejecutar la aplicación como uno o más procesos sin estado

## **VII. Asignación de puertos**

Publicar servicios mediante asignación de puertos

## **VIII. Concurrencia**

Escalar mediante el modelo de procesos

## **IX. Disponibilidad**

Hacer el sistema más robusto intentando conseguir inicios rápidos y finalizaciones seguras

## **X. Paridad en desarrollo y producción**

Mantener desarrollo, preproducción y producción tan parecidos como sea posible

## **XI. Historiales**

Tratar los historiales como una transmisión de eventos

## **XII. Administración de procesos**

Ejecutar las tareas de gestión/administración como procesos que solo se ejecutan una vez



# Jugando con Estilos y Patrones

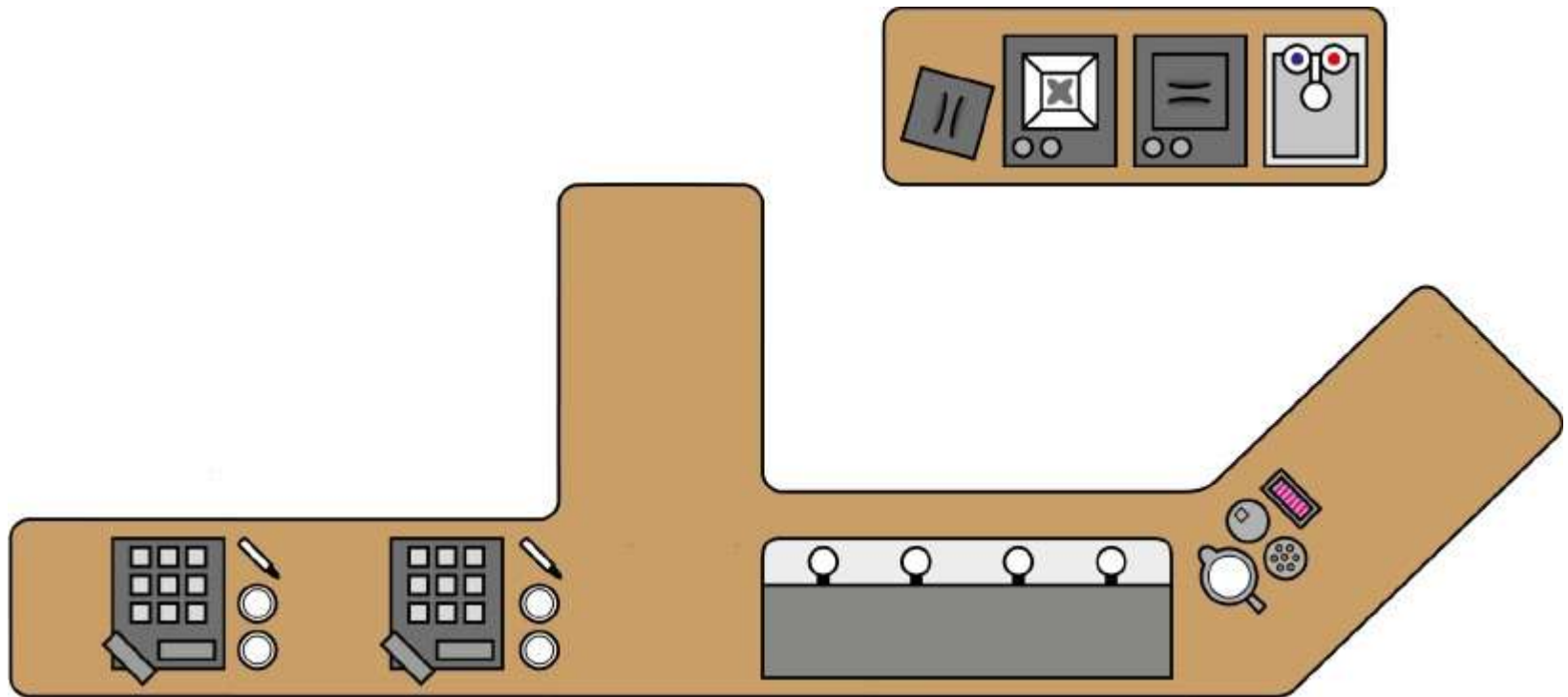




¿ Cual es el DRIVER de negocio ?

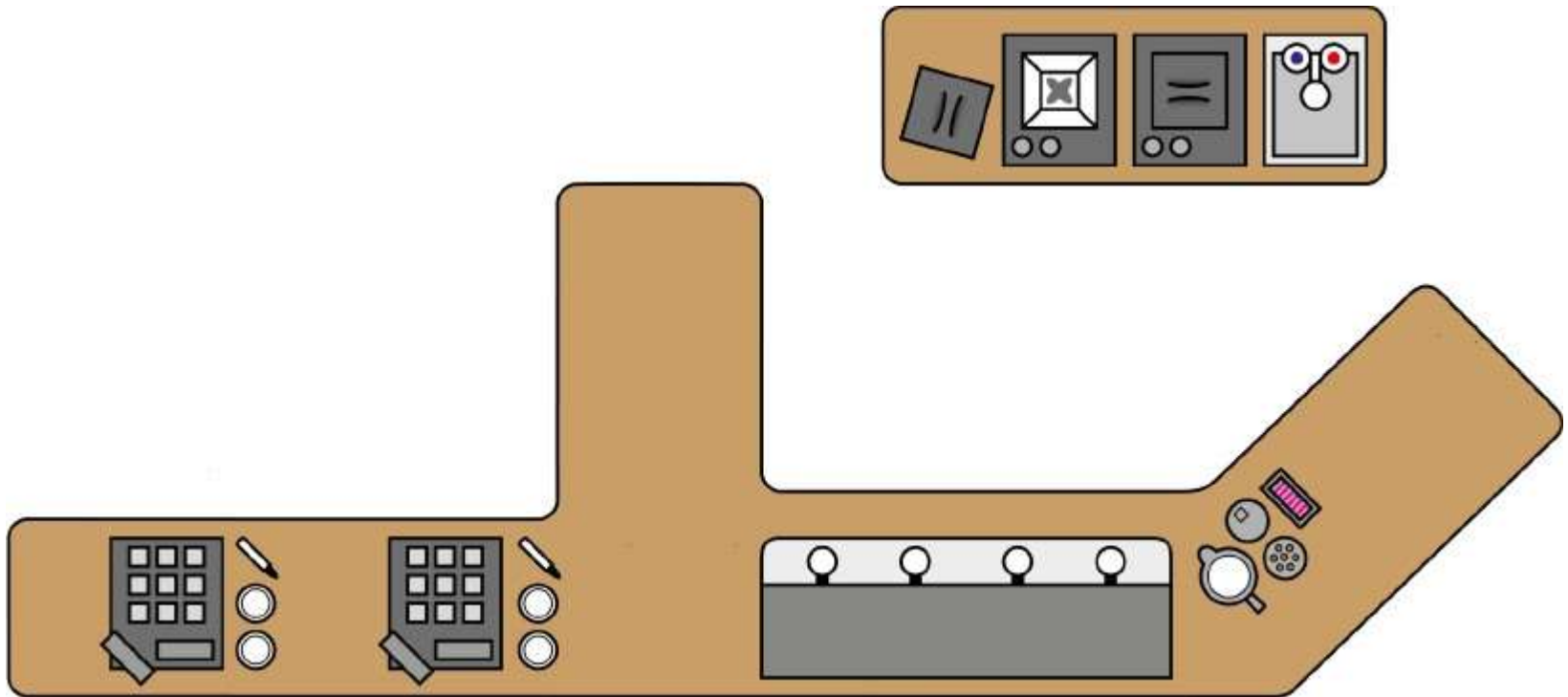
¿ Cual es el DRIVER de arquitectura ?

# Construye la Aplicación



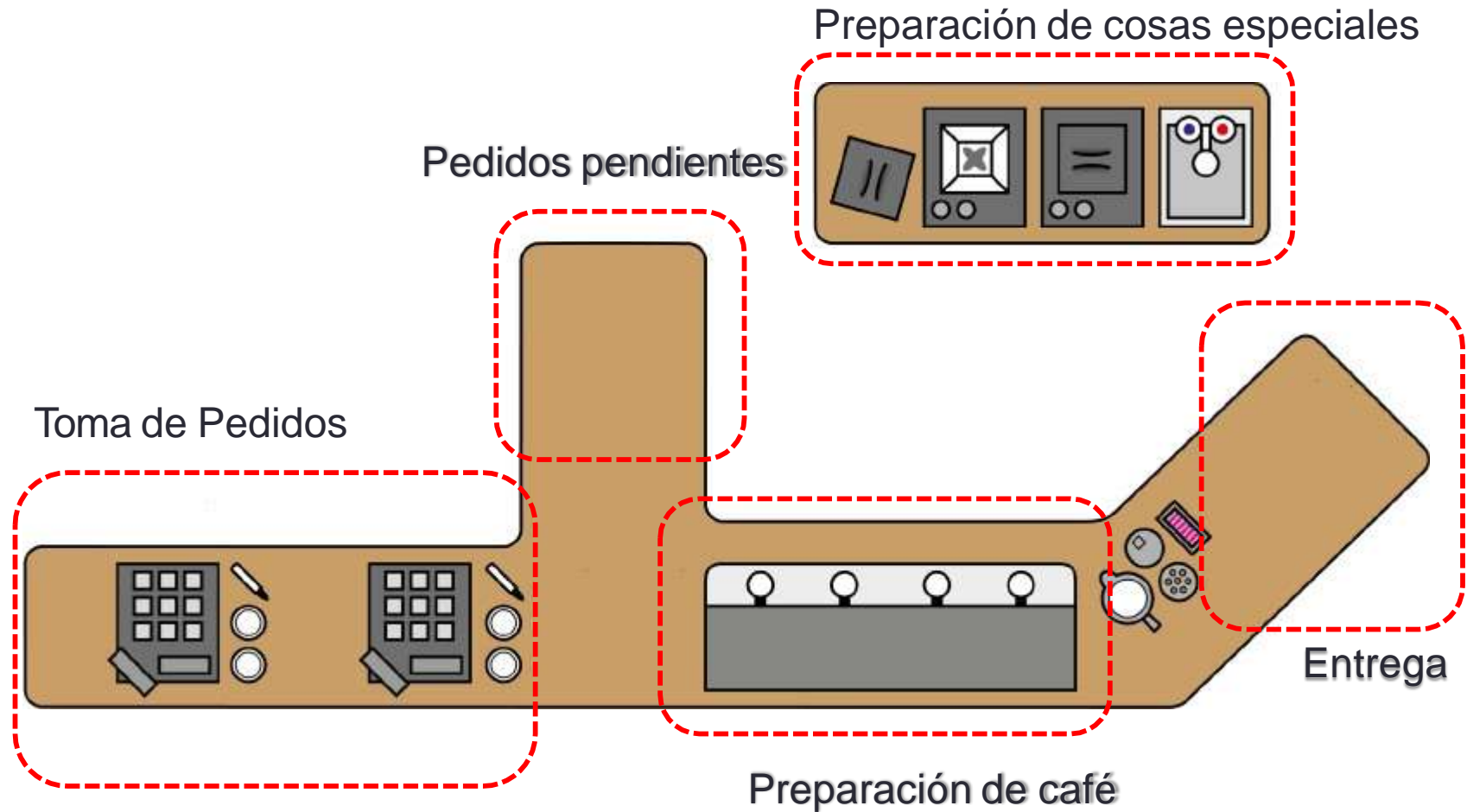
## Descomponer la funcionalidad

- Cohesión
- Acoplamiento



## Descomponer la funcionalidad

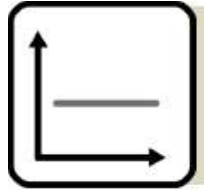
- Cohesión
- Acoplamiento



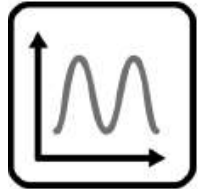
¿ qué estilo /patrón de arquitectura emplear ?  
¿ por qué ?



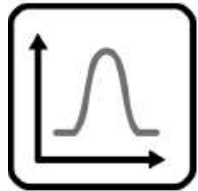
## Para confirmar el Estilo/patrón ....



Estático



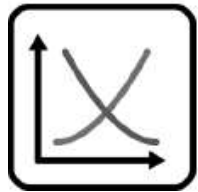
Periódico



Pico único



Aleatorio



Continuo

Deberíamos analizar la carga de cada componente del sistema

Deberíamos analizar los recursos de cada componente del sistema



Preparación de cosas especiales

Pedidos pendientes

Toma de Pedidos

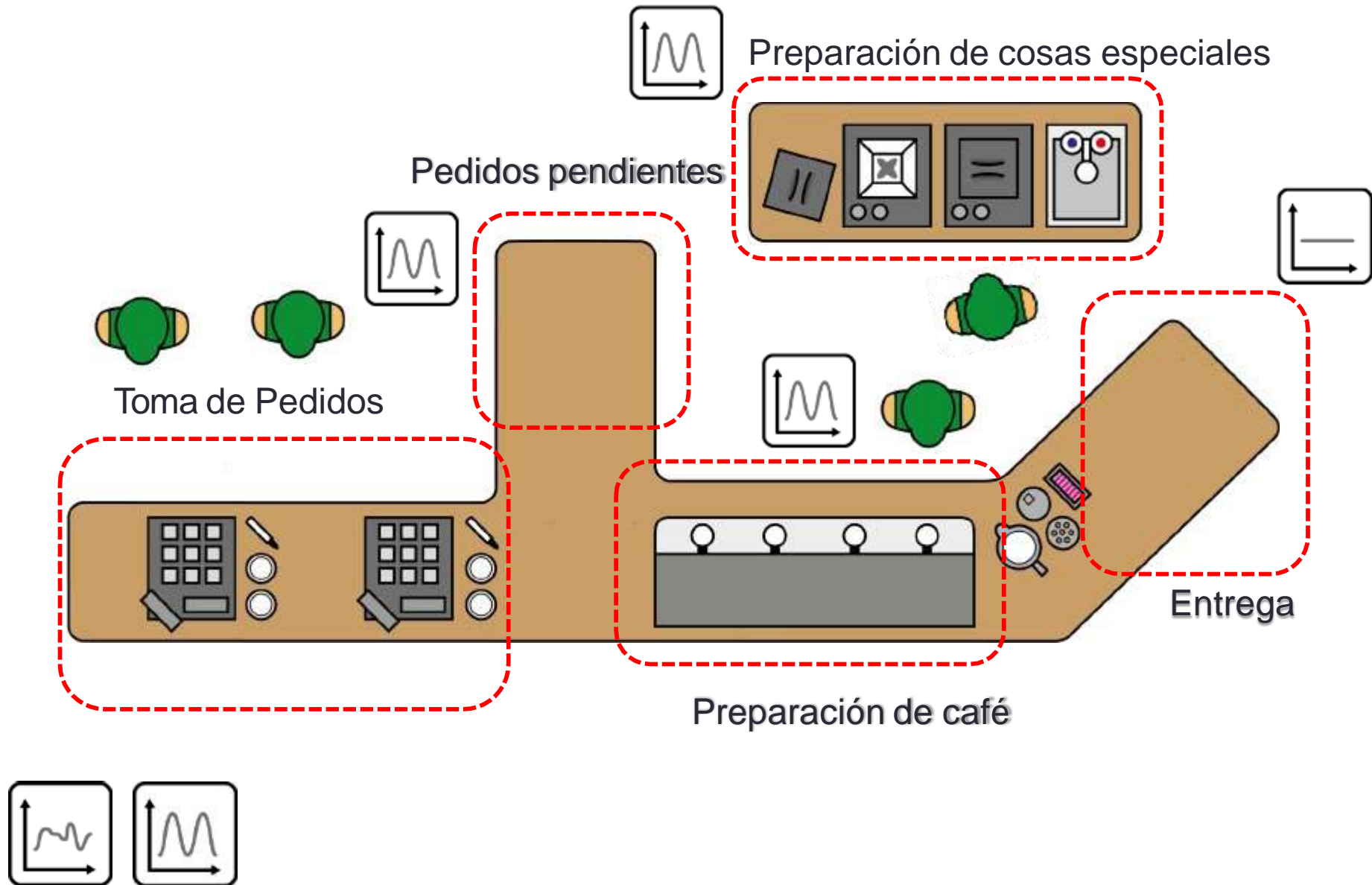
Entrega

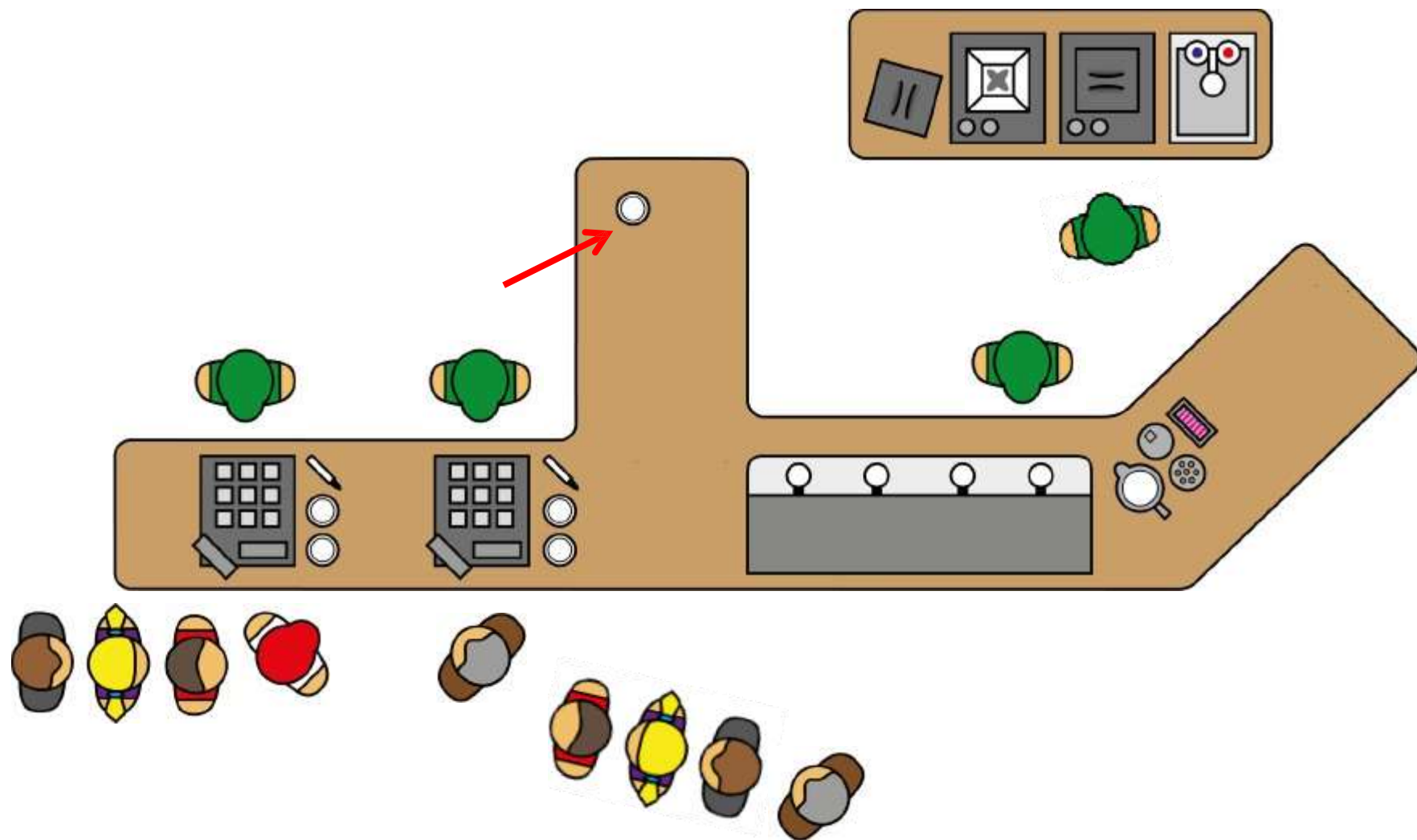
Preparación de café

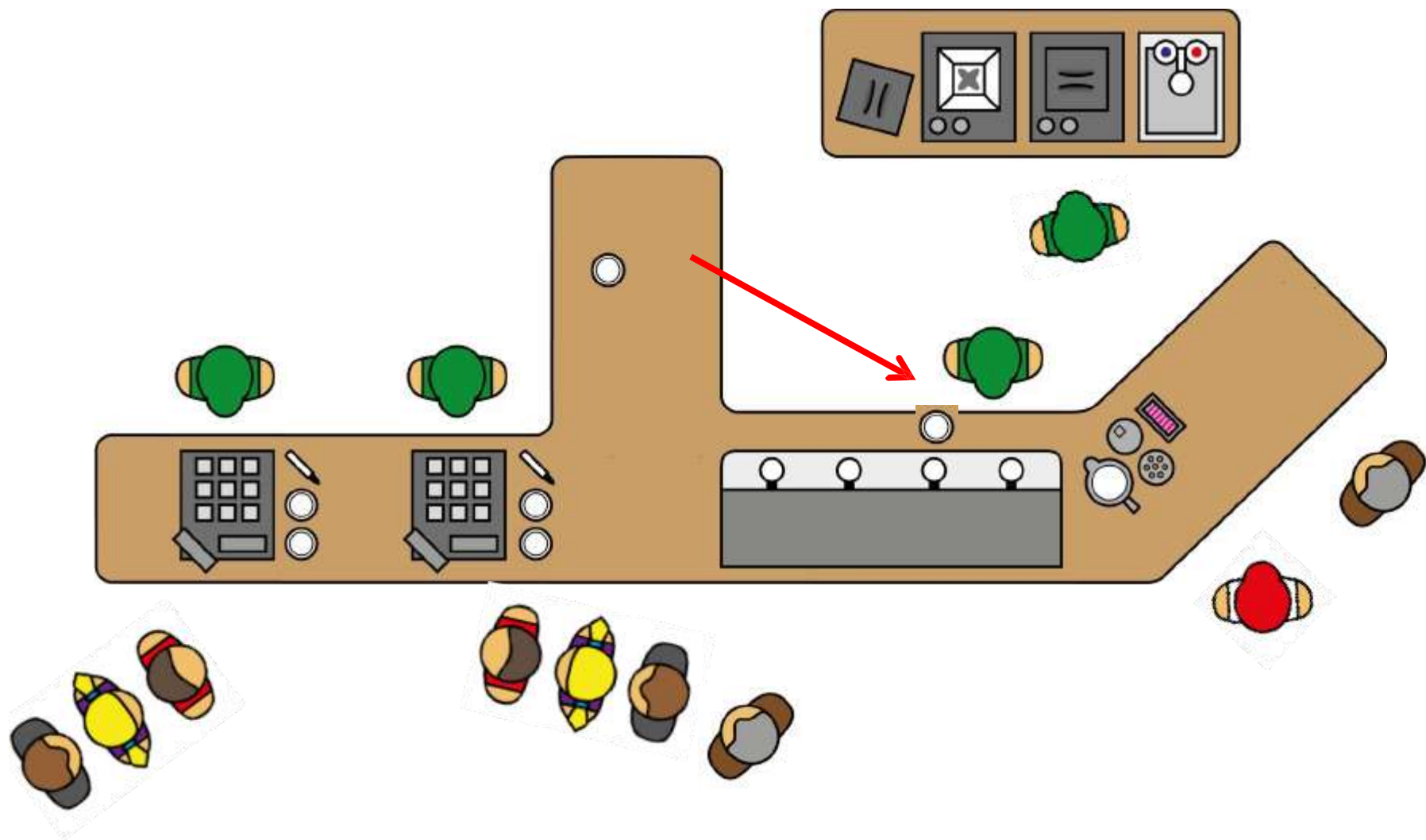


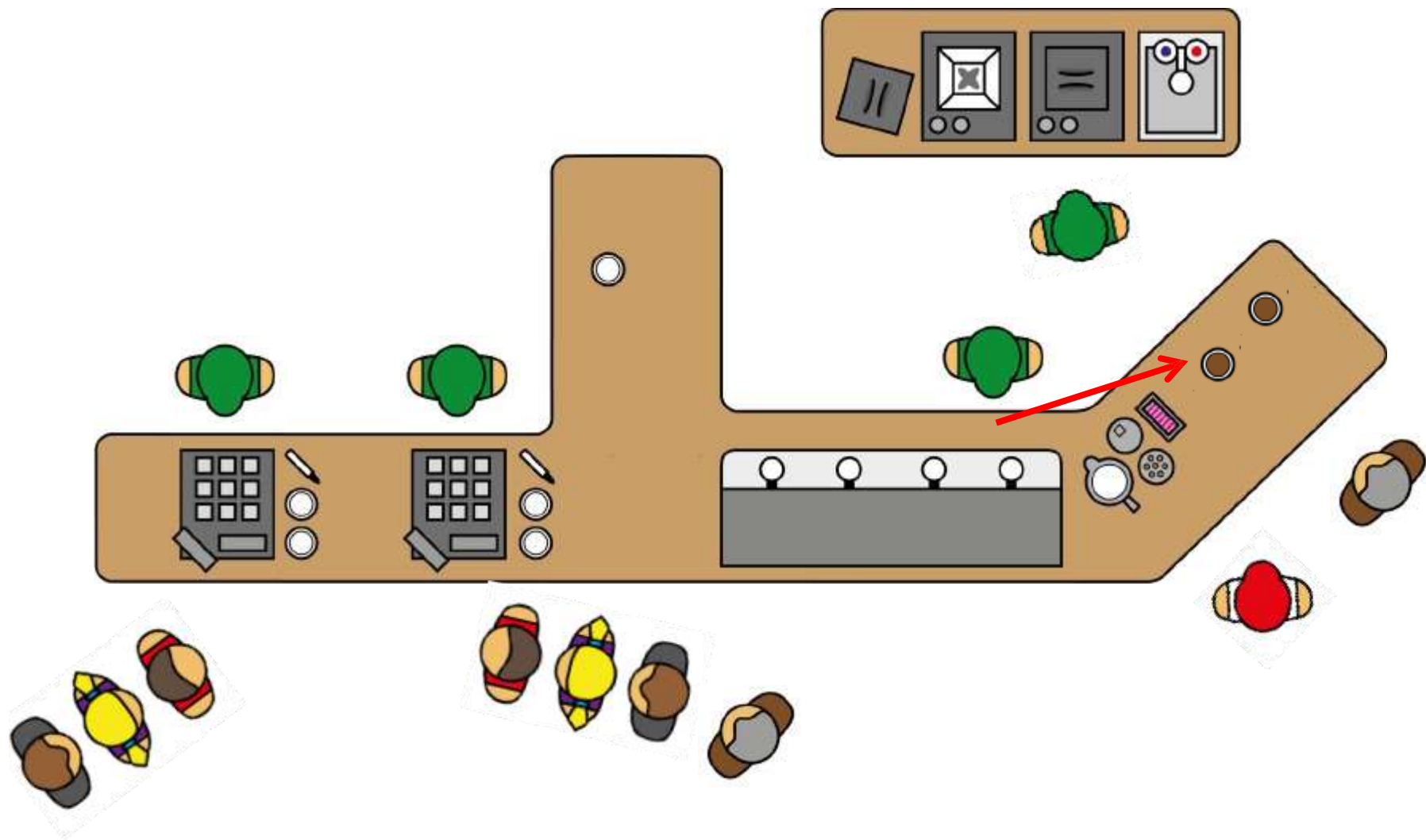


## Carga del Sistema : **WORKLOAD**









# Construye la Aplicación

