***Ejercicio 1.*** *A partir de los siguientes fragmentos de código, identifique problemas y proponga soluciones.*

**a.**

```
struct Producer {
    buffer: Arc<(Condvar, Mutex<Vec<i32>>)>,
}
impl Actor for Producer {
    type Context = Context<Self>;
}
impl Handler<Produce> for Producer {
    type Result = ();
    fn handle(&mut self, msg: Produce, _ctx: &mut Context<Self>) -> Self::Result {
        let (cvar, lock) = &*self.buffer;
        let mut guard = cvar.wait_while(lock.lock().unwrap(), |buf|{ buf.len() >= 10 })
        *guard.push(msg.value)
    }
}
impl Handler<Consume> for Producer {
    type Result = ();
    fn handle(&mut self, _msg: Consume, _ctx: &mut Context<Self>) -> Self::Result {
        let (cvar, lock) = &*self.buffer;
        let mut guard = cvar.wait_while(lock.lock().unwrap(), |buf|{ buf.len() == 0 })
        return *guard.pop();
    }
}
```

**b.**

```
impl Handler<PrepararCafe> for Barista {
    fn handle(&mut self, _msg: PrepararCafe, _ctx: &mut Context<Self>) {
        loop {
            for maquina in self.maquinas.iter() {
                match maquina.send(EstasLibre).await {
                    Ok(true) => {
                        println!("Barista. Máquina libre, preparando café");
                        // Preparar café (sin esperar resultado)
                        maquina.do_send(PrepararCafe);
                        return;
                    },
                    Ok(false) => {
                        println!("Barista. Sigo buscando máquina libre");
                    },
                    Err(e) => {
                        println!("Barista. Error: {}", e);
                    }
                }
            }
        }
    }
}
```

**Ejercicio 2**

Verdadero o Falso. Justifique

a. En un ambiente de ejecución con una única CPU, un conjunto de hilos de procesamiento intensivo tomarán un tiempo de ejecución significativamente menor a un conjunto de tareas asincrónicas que ejecuten el mismo procesamiento.
b. Programando en el modelo fork-join se debe tener especial cuidado en evitar condiciones de carrera.
c. En el estado mutable compartido los hilos deben verificar periódicamente por actualizaciones de dicho estado
d. El estado interno de un actor se encuentra protegido por una exclusión mutua.

**Ejercicio 3**

Describa y justifique con que modelo de concurrencia modelaría la implementación para cada uno de los siguientes casos de uso

- Un sistema de subastas en línea donde múltiples usuarios ofrecen por ítems en tiempo real, se debe actualizar el precio del producto y notificar a los usuarios involucrados.
- Una aplicación de una pinturería que aplica filtros en una imagen para ver cómo quedarían distintos colores sobre una pared.
- Una memoria cache utilizada por un modelo de Machine Learning usado para reconocimiento facial.
- Una herramienta que dado N imágenes (donde N es muy grande) las junta en grupos de 100 y las transforma a PDF.

**Ejercicio 4**

En una oficina sobre la 9 de Julio trabaja una empresa que posee una serie de máquinas de café automáticas distribuidas en todas las áreas, a lo largo de los 10 pisos que alquila.

Para comprar café en las máquinas, cada empleado tiene una tarjeta cargada con créditos. Sin embargo, los créditos son compartidos por equipo para mejor control de cada gerencia.

Los créditos se cargan el primer día hábil de cada mes, y además cada gerente tiene la posibilidad de agregar montos extra a lo largo del mes como bonus cuando considera que sus empleados deben ser recompensados.

Modele el sistema con actores, describiendo sus estados y mensajes en pseudocódigo de Rust. Describa cómo reaccionan ante los mensajes que reciben (en prosa).

Especifique cómo asegura su modelo el caso del "doble gasto" si dos tarjetas asociadas al mismo equipo se utilizan en dos máquinas al mismo tiempo.