

# Cambios Realizados - Sistema de Temporizadores

---

## Fecha

---

8 de noviembre de 2025

## Resumen de Modificaciones

---

### 1. Eliminación de FloatingTimer

- **✓ Eliminado:** components/modals/FloatingTimer.tsx
- **✓ Removido import** en app/page.tsx
- **✓ Removido uso del componente** en app/page.tsx
- **✓ Removida función** handleRefreshCounter que solo era usada por FloatingTimer
- **✓ Removidas referencias** en hooks/useTimer.ts :
- Estado showFloatingTimer
- Función setShowFloatingTimer
- Todas las llamadas a setShowFloatingTimer()

### 2. Configuración de Tiempo Centralizada

**Archivo:** lib/economyConfig.ts

Agregado nueva configuración:

```
// Refresh button eligibility configuration
// Buttons will be enabled when timer <= this value (in seconds)
refreshButtonEligibility: {
  timeGrantedByLevel: 300, // 5 minutes in seconds - configurable time threshold
}
```

#### Beneficios:

- Valor configurable centralizado
- Fácil de cambiar en el futuro
- Un solo lugar para modificar el tiempo otorgado por nivel

### 3. Lógica de Deshabilitación de Botones

**TopNavigation.tsx**

**Condición de deshabilitación:**

```
disabled={
  userPoints < 40 ||
  isPageBlocked ||
  isRefreshingTopbar ||
  timer > ECONOMY.refreshButtonEligibility.timeGrantedByLevel
}
```

### **Tooltip dinámico:**

- Si `timer > tiempo_otorgado` : "Solo puedes refrescar cuando el tiempo sea menor a 5 minutos"
- Si `userPoints < 40` : "Necesitas 40 puntos para refrescar"
- De lo contrario: "Refrescar 40 puntos"

### **RefreshCounterButton.tsx**

#### **Condición de deshabilitación:**

```
const isTimeEligible = timeLeftSeconds <= EC0-
NOMY.refreshButtonEligibility.timeGrantedByLevel
const isButtonEnabled = userPoints >= 40 && isTimeEligible && !loading
```

### **Tooltip dinámico:**

- Calcula automáticamente los minutos desde la configuración
- Mensajes adaptativos según el estado

## **4. Funcionalidad Preservada**

### **ContadorUsuario.tsx**

- Mantiene toda su funcionalidad original
- Muestra el temporizador correctamente
- Botones de "Extender Tiempo" e "Información" funcionando
- Integración con `RefreshCounterButton` actualizada

### **TopNavigation.tsx**

- Todos los contadores funcionando
- Navegación intacta
- Menú de usuario preservado
- Notificaciones funcionando

## **Fórmula de Deshabilitación**

### **Antes (Incorrecto)**

Los 40 puntos se consideraban parte de la condición de deshabilitación.

### **Ahora (Correcto)**

```
Si tiempo_actual > tiempo_otorgado_por_nivel (300s) → Botones DESHABILITADOS
Si tiempo_actual <= tiempo_otorgado_por_nivel (300s) → Botones HABILITADOS
```

**Importante:** Los 40 puntos son el COSTO de la operación, NO parte de la condición de deshabilitación.

## Configuración Actual

| Parámetro                 | Valor                    | Ubicación            |
|---------------------------|--------------------------|----------------------|
| Tiempo otorgado por nivel | 300 segundos (5 minutos) | lib/economyConfig.ts |
| Costo de refresh          | 40 puntos                | lib/economyConfig.ts |

## Cómo Cambiar el Tiempo en el Futuro

Para cambiar el tiempo otorgado por nivel, editar `lib/economyConfig.ts`:

```
refreshButtonEligibility: {
  timeGrantedByLevel: 600, // Por ejemplo, 10 minutos
}
```

Todos los botones se actualizarán automáticamente.

## Archivos Modificados

1.  `lib/economyConfig.ts` - Configuración centralizada
2.  `components/TopNavigation.tsx` - Lógica de deshabilitación
3.  `components/RefreshCounterButton.tsx` - Lógica de deshabilitación
4.  `app/page.tsx` - Eliminación de FloatingTimer
5.  `hooks/useTimer.ts` - Eliminación de referencias
6.  `components/modals/FloatingTimer.tsx` - ELIMINADO

## Verificación

- FloatingTimer completamente eliminado
- No hay referencias rotas
- Configuración centralizada implementada
- Lógica de deshabilitación correcta
- Tooltips dinámicos
- Funcionalidad de otros contadores preservada