

Testing Instructions: Automatic Totem Consumption Feature

Overview

This document provides step-by-step instructions for testing the automatic totem consumption feature when a user's counter expires.

Prerequisites

1. Database Setup

Before testing, ensure the database is up-to-date with the new schema:

```
# Run Prisma migrations to create the TotemConsumption table
npx prisma migrate dev

# Or in production
npx prisma migrate deploy

# Generate Prisma Client
npx prisma generate
```

2. Environment Configuration

Set up the `MAX_AUTO_TOTEMS_PER_DAY` environment variable in your `.env` file:

```
# For unlimited auto-consumption (default)
MAX_AUTO_TOTEMS_PER_DAY=0

# For testing daily limits (e.g., 3 totems per day)
MAX_AUTO_TOTEMS_PER_DAY=3
```

3. Application Setup

```
# Install dependencies (if not already done)
npm install

# Start the development server
npm run dev
```

The application should be running at `http://localhost:3000`

Test Scenarios

Test Case 1: User with Available Totems (Auto-consumption Success)

Objective: Verify that a totem is automatically consumed when the counter expires and the user has totems available.

Setup:

1. Create or select a test user with the following attributes:

- `totems >= 1`
- `counterExpiresAt` set to expire soon or already expired
- `isSuspended = false`

Using Prisma Studio:

```
npx prisma studio
```

Navigate to the User model and update a test user.

Using SQL:

```
UPDATE "User"
SET totems = 2,
    "counterExpiresAt" = NOW() - INTERVAL '1 minute',
    "isSuspended" = false
WHERE email = 'testuser@example.com';
```

Test Steps:

1. Log in as the test user
2. Navigate to the main page
3. Wait for the counter to reach 0 or refresh the page if already expired

4. Expected Results:

- A toast notification appears with: “Se usó 1 tótem y tu contador se reactivó a 5:00.”
- The counter resets to 5:00 (5 minutes)
- The totem count decreases by 1
- The suspension window does NOT appear
- User points remain unchanged
- User can continue using the application normally

Verification:

Check the database to confirm the audit trail:

```
SELECT * FROM "TotemConsumption"
WHERE "userId" = '<user-id>'
ORDER BY "createdAt" DESC
LIMIT 1;
```

Expected fields:

- `type = 'auto'`
- `reason = 'suspend_resume'`
- `delta = -1`

- `balanceBefore` = (previous totem count)
 - `balanceAfter` = (previous totem count - 1)
 - `previousExpiresAt` = (old expiration time)
 - `newExpiresAt` = (new expiration time, ~5 minutes from now)
-

Test Case 2: User with No Totems (Suspension)

Objective: Verify that the suspension window appears when the counter expires and the user has no totems.

Setup:

1. Update the test user:

- `totems` = 0
- `counterExpiresAt` set to expire soon or already expired
- `isSuspended` = false

Using SQL:

```
UPDATE "User"
SET totems = 0,
    "counterExpiresAt" = NOW() - INTERVAL '1 minute',
    "isSuspended" = false
WHERE email = 'testuser@example.com';
```

Test Steps:

1. Log in as the test user
2. Navigate to the main page
3. Wait for the counter to reach 0 or refresh if already expired

Expected Results:

- NO toast notification appears
- The “Cuenta Suspendida” (Account Suspended) window appears
- The user’s `isSuspended` field is set to true in the database
- The user cannot access other tabs/features
- A notification record is created with type ‘suspended_for_counter’

Verification:

```
-- Check user suspension status
SELECT "isSuspended", "suspendedAt", totems
FROM "User"
WHERE email = 'testuser@example.com';

-- Check notification
SELECT * FROM "Notification"
WHERE "userId" = '<user-id>'
    AND type = 'suspended_for_counter'
ORDER BY "createdAt" DESC
LIMIT 1;
```

Test Case 3: Concurrent Requests (Idempotency Test)

Objective: Verify that only one totem is consumed even if multiple requests occur simultaneously.

Setup:

1. Update the test user:

```
- totems = 3
- counterExpiresAt = expired
- isSuspended = false
```

Test Steps:

1. Open the application in **3 different browser tabs** simultaneously
2. Log in as the same test user in all tabs
3. Ensure the counter is at 0 in all tabs
4. All tabs should trigger the heartbeat counter check simultaneously

5. Expected Results:

- Only ONE totem is consumed (user should have 2 totems remaining)
- Only ONE toast notification appears (may show in multiple tabs)
- Only ONE audit log entry is created for this consumption event
- All tabs sync to show the updated counter and totem count

Verification:

```
-- Check totem consumption audit logs
SELECT * FROM "TotemConsumption"
WHERE "userId" = '<user-id>'
    AND "createdAt" > NOW() - INTERVAL '1 minute'
ORDER BY "createdAt" DESC;

-- Should only show 1 record for the concurrent requests

-- Check user's current totem count
SELECT totems FROM "User" WHERE id = '<user-id>';
-- Should be 2 (started with 3, consumed 1)
```

Test Case 4: Daily Limit Reached (Limit Enforcement)

Objective: Verify that when the daily limit is reached, no more totems are auto-consumed and the suspension window appears.

Setup:

1. Set the environment variable:

```
MAX_AUTO_TOTEMS_PER_DAY=2
```

1. Create test data for today's consumption:

```
-- Simulate 2 totem consumptions today
INSERT INTO "TotemConsumption"
    ("id", "userId", "type", "reason", "delta", "balanceBefore", "balanceAfter", "previousExpiresAt", "newExpiresAt", "createdAt")
VALUES
    (gen_random_uuid(), '<user-id>', 'auto', 'suspend_resume', -1, 3, 2, NOW() - INTERVAL '2 hours', NOW() - INTERVAL '2 hours' + INTERVAL '5 minutes', NOW() - INTERVAL '2 hours'),
    (gen_random_uuid(), '<user-id>', 'auto', 'suspend_resume', -1, 2, 1, NOW() - INTERVAL '1 hour', NOW() - INTERVAL '1 hour' + INTERVAL '5 minutes', NOW() - INTERVAL '1 hour');

-- Update user to have totems available but counter expired
UPDATE "User"
SET totems = 5,
    "counterExpiresAt" = NOW() - INTERVAL '1 minute',
    "isSuspended" = false
WHERE id = '<user-id>';
```

Test Steps:

1. Log in as the test user
2. Navigate to the main page
3. Wait for counter to reach 0

4. Expected Results:

- NO totem is consumed (daily limit reached)
- NO toast notification appears
- The “Cuenta Suspendida” window appears
- User’s `isSuspended` is set to true
- User’s totem count remains unchanged (still 5)
- Notification includes note about ‘`daily_totem_limit_reached`’

Verification:

```
-- Count today's auto-consumptions
SELECT COUNT(*) as today_consumptions
FROM "TotemConsumption"
WHERE "userId" = '<user-id>'
    AND type = 'auto'
    AND "createdAt" >= CURRENT_DATE;
-- Should be 2 (not 3)

-- Check user status
SELECT totems, "isSuspended" FROM "User" WHERE id = '<user-id>';
-- totems should still be 5, isSuspended should be true
```

Test Case 5: Counter Not Expired (No Action)

Objective: Verify that nothing happens when the counter is still active.

Setup:

1. Update the test user:

```
- totems = 2
```

- `counterExpiresAt` = future date (e.g., +4 minutes)
- `isSuspended` = false

```
UPDATE "User"
SET totems = 2,
    "counterExpiresAt" = NOW() + INTERVAL '4 minutes',
    "isSuspended" = false
WHERE email = 'testuser@example.com';
```

Test Steps:

1. Log in as the test user
2. Navigate to the main page
3. Observe the counter (should show ~4:00 remaining)

4. Expected Results:

- Counter displays the remaining time normally
- NO toast notification appears
- NO totem is consumed
- NO suspension window appears
- User can use all features normally

Database Queries for Testing

Query All Totem Consumptions for a User

```
SELECT
    "id",
    "type",
    "reason",
    "delta",
    "balanceBefore",
    "balanceAfter",
    "previousExpiresAt",
    "newExpiresAt",
    "createdAt"
FROM "TotemConsumption"
WHERE "userId" = '<user-id>'
ORDER BY "createdAt" DESC;
```

Check Daily Consumption Count

```
SELECT COUNT(*) as consumptions_today
FROM "TotemConsumption"
WHERE "userId" = '<user-id>'
    AND "type" = 'auto'
    AND "createdAt" >= CURRENT_DATE;
```

Reset User for New Test

```
UPDATE "User"
SET
  totems = 3,
  "counterExpiresAt" = NOW() + INTERVAL '5 minutes',
  "isSuspended" = false,
  "suspendedAt" = NULL
WHERE email = 'testuser@example.com';

-- Optionally clear test audit logs
DELETE FROM "TotemConsumption"
WHERE "userId" = '<user-id>' AND "createdAt" > NOW() - INTERVAL '1 day';
```

Manual Testing Checklist

- [] Test Case 1: Auto-consumption with available totems ✓
 - [] Test Case 2: Suspension with no totems ✓
 - [] Test Case 3: Concurrent requests (idempotency) ✓
 - [] Test Case 4: Daily limit enforcement ✓
 - [] Test Case 5: Counter not expired (no action) ✓
 - [] Verify toast notification message is correct
 - [] Verify counter resets to exactly 5:00 (300 seconds)
 - [] Verify points balance remains unchanged after totem consumption
 - [] Verify audit logs are created correctly
 - [] Verify user can continue normal operations after totem consumption
 - [] Verify suspension window only appears when appropriate
 - [] Test across different user ranks (registrado, invitado, basico, vip, premium, elite)
 - [] Test with VIP/Premium/Elite users who have base totems (totem floor feature)
-

Troubleshooting

Issue: Totem not being consumed automatically

Possible causes:

- Database migration not run (`npx prisma migrate deploy`)
- Backend not restarted after code changes
- UserId not being passed correctly to ContadorUsuario component
- Timer state not updating properly

Debug steps:

1. Check browser console for errors
2. Check server logs for backend errors
3. Verify userId is present: `console.log(userId)` in page.tsx
4. Test heartbeatCounter directly:

`typescript`

```
import { heartbeatCounter } from '@/app/actions/counter'
```

```
const result = await heartbeatCounter('user-id-here')
console.log(result)
```

Issue: Toast notification not appearing

Possible causes:

- react-hot-toast not installed properly
- Toaster component not in layout
- Toast message being blocked by browser

Debug steps:

1. Verify installation: `npm list react-hot-toast`
2. Check if Toaster is in app/layout.tsx
3. Try showing a test toast: `toast.success('Test')` in browser console

Issue: Daily limit not working

Possible causes:

- MAX_AUTO_TOTEMS_PER_DAY environment variable not set
- Environment variable not loaded (restart needed)
- Test data for today's consumptions not set up correctly

Debug steps:

1. Check env variable: `console.log(process.env.MAX_AUTO_TOTEMS_PER_DAY)`
 2. Restart the dev server: `npm run dev`
 3. Verify count query returns correct number
-

Production Deployment Checklist

Before deploying to production:

- [] Run database migrations: `npx prisma migrate deploy`
 - [] Set MAX_AUTO_TOTEMS_PER_DAY in production environment variables
 - [] Test all scenarios in staging environment first
 - [] Monitor audit logs for the first few days after deployment
 - [] Set up alerts for unusual totem consumption patterns
 - [] Document the feature for end users
 - [] Train support team on the new feature
-

Expected Behavior Summary

Scenario	Totems Available	Daily Limit	Result
Counter expired	Yes (≥ 1)	Not reached	Auto-consume 1 totem, reset counter, show toast
Counter expired	Yes (≥ 1)	Reached	Suspend user, show suspension window
Counter expired	No (0)	N/A	Suspend user, show suspension window
Counter active	Any	N/A	No action, continue normally

Support & Questions

If you encounter any issues during testing:

1. Check the browser console for client-side errors
2. Check server logs for backend errors
3. Verify database schema matches expected structure
4. Ensure all dependencies are installed correctly
5. Review the implementation files for any recent changes

For additional assistance, refer to the implementation files:

- Backend: `app/actions/counter.ts`
- Frontend: `components/ContadorUsuario.tsx` , `app/page.tsx`
- Schema: `prisma/schema.prisma`
- Config: `.env.example` , `lib/economyConfig.ts`