






YigiCoin Refresh Feature - Implementation Guide

Quick Start

What's New?

Users can now refresh their account counter by spending 10 points, which resets their timer to the full duration for their current rank.


Key Features

1.  Centralized rank duration configuration
 2.  Two refresh buttons in the UI (header and panel)
 3.  Point validation (requires 10 points)
 4.  Success/error feedback messages
 5.  API endpoints for refresh and testing
-

Refresh Button Locations


1. Top Navigation Bar

Located in the header next to “Tiempo restante” timer

- **Icon:**  (Refresh icon)
- **Color:** Green when active, gray when disabled
- **Position:** Between info button and notifications


2. Estado de la Cuenta Panel

Located next to the large countdown timer

- **Icon:**  (Refresh icon)
 - **Color:** White with blue icon
 - **Position:** Right side of the timer display
-

How It Works

User Flow

1. User clicks refresh button (requires 10 points)
2. System validates point balance
3. If sufficient:
 - Deducts 10 points
 - Resets counter to rank duration
 - Shows success message: “Contador reiniciado  (-10 pts)”
 - Page auto-reloads to update timer

4. If insufficient:

- Shows error: "Puntos insuficientes, asciende o mira anuncios para conseguir puntos extras"
- Button remains disabled

Rank Durations

Rank	Duration	Seconds
Registrado	48 hours	172,800
Invitado	72 hours	259,200
Básico	96 hours	345,600
VIP	120 hours	432,000
Premium	168 hours	604,800
Élite	240 hours	864,000

Technical Implementation

Files Modified/Created

New Files

- `/lib/counter.ts` - Centralized duration config & utilities
- `/hooks/useRefresh.ts` - React hook for refresh logic
- `/app/api/refresh/route.ts` - POST endpoint for refresh
- `/app/api/invites/dev-conversion/route.ts` - Dev testing endpoint
- `/CHANGELOG.md` - Detailed change documentation

Modified Files

- `/hooks/useSimulation.ts` - Updated to use RANK_DURATIONS
- `/components/TopNavigation.tsx` - Added header refresh button
- `/components/ContadorUsuario.tsx` - Added panel refresh button

API Endpoints

POST /api/refresh

Refreshes user's counter by deducting 10 points.

Request:

```
{
  "userId": "string"
}
```

Success Response:

```
{
  "ok": true,
  "user": { ...userData },
  "message": "Contador reiniciado ✅ (-10 pts)"
}
```

Error Response (Insufficient Points):

```
{
  "ok": false,
  "code": "INSUFFICIENT_POINTS",
  "message": "Puntos insuficientes..."
}
```

POST /api/invites/dev-conversion

Development endpoint for testing invite conversion counter reset.

Request:

```
{
  "inviterId": "string"
}
```

Response:

```
{
  "ok": true,
  "user": { ...userData },
  "message": "Counter reset for inviter..."
}
```

Note: Only works for users with rank === 'Registrado'

Testing

Manual Testing Steps

1. Test with Sufficient Points:

...

2. Set user points to 15 (in localStorage)
3. Click refresh button
4. Verify: Points reduced to 5
5. Verify: Counter reset to rank duration
6. Verify: Success message displays

...

7. Test with Insufficient Points:

...

8. Set user points to 5

9. Verify: Refresh button is disabled
10. Hover over button
11. Verify: Error message displays

12. Test Invite Conversion:

```
POST /api/invites/dev-conversion
Body: { "inviterId": "test123" }
```

Expected: Counter resets to 48h for Registrado users

Test Data Setup

```
// Set user points in browser console
const userData = JSON.parse(localStorage.getItem('user_simulation_data') || '{}');
userData.points = 15; // Set desired points
localStorage.setItem('user_simulation_data', JSON.stringify(userData));
location.reload(); // Refresh page
```

UI States

Button States

1. **Active** (Green/White):
 - User has ≥ 10 points
 - Page not blocked
 - Not currently refreshing
2. **Disabled** (Gray):
 - User has < 10 points
 - Page is blocked
 - Refresh in progress
3. **Loading** (Spinning):
 - Refresh request in progress
 - Icon animates with spin

Messages

- **Success:** Green background, 3-second auto-dismiss
- **Error:** Red background, persists until dismissed
- **Tooltip:** Dark gray, shows on hover

Data Storage

localStorage Keys

- `user_simulation_data` - Main user data

- `points` - User's point balance
- `currentRank` - User's current rank
- `counterEndsAt` - ISO timestamp when counter expires
- `lastRefresh` - ISO timestamp of last refresh

Data Updates on Refresh

```
{
  points: currentPoints - 10,
  counterEndsAt: new Date(now + rankDuration * 1000).toISOString(),
  lastRefresh: new Date().toISOString()
}
```

Deployment

Prerequisites

- Node.js >= 22
- npm >= 10
- Next.js 14

Build Commands

```
# Install dependencies
npm install

# Development mode
npm run dev

# Production build
npm run build

# Start production server
npm run start
```

Environment

No environment variables required - this feature uses localStorage only.

Security Notes

1. **Client-Side Validation:** All validation happens in localStorage (simulation mode)
 2. **Production Requirements:**
 - Implement server-side validation
 - Add authentication middleware
 - Store points in database
 - Add transaction logging
 - Implement rate limiting
-

Troubleshooting

Issue: Button is disabled

Solution: Check user points in localStorage

```
const userData = JSON.parse(localStorage.getItem('user_simulation_data'));
console.log('Points:', userData.points); // Should be >= 10
```

Issue: Refresh doesn't work

Solution: Check console for errors and verify localStorage data

```
// Verify data structure
const userData = JSON.parse(localStorage.getItem('user_simulation_data'));
console.log('User Data:', userData);
```

Issue: Counter doesn't reset

Solution: Ensure page reloads after refresh

```
// Manual refresh trigger
window.location.reload();
```

Support

For issues or questions:

1. Check CHANGELOG.md for detailed changes
 2. Review browser console for error messages
 3. Verify localStorage data integrity
 4. Ensure user has sufficient points (≥ 10)
-

Feature Checklist

- [x] Centralized rank duration configuration
- [x] POST /api/refresh endpoint
- [x] POST /api/invites/dev-conversion endpoint
- [x] useRefresh custom hook
- [x] Header refresh button
- [x] Panel refresh button
- [x] Point validation
- [x] Success/error messages
- [x] Responsive design
- [x] Loading states
- [x] Disabled states
- [x] Auto-reload after refresh

- [x] Comprehensive documentation
- [x] Git version control

Version: 1.1.0

Date: October 21, 2025

Project: YigiCoin Platform

Feature: Counter Refresh with Points System

END OF GUIDE