



Yega Dashboard - Ecosistema GitHub

Dashboard neón para monitorear y analizar el ecosistema completo de repositorios Yega en GitHub.



Características

- ✨ **Diseño Neón:** Interfaz con fondo negro y colores neón (rosa, cian, morado, verde manzana)
- 📈 **Monitoreo en Tiempo Real:** Sincronización automática con GitHub API
- 📁 **Multi-Repositorio:** Seguimiento de los 5 componentes del ecosistema Yega
- 🐛 **Gestión de Issues:** Vista completa de issues y pull requests
- 📊 **Visualizaciones:** Gráficos interactivos con Chart.js
- 📖 **README Viewer:** Visualización de documentación en modal



Repositorios Monitoreados

1. **Yega-Ecosistema** - Core del ecosistema
2. **Yega-API** - Backend y servicios API
3. **Yega-Tienda** - Aplicación de comercio
4. **Yega-Repartidor** - App para delivery
5. **Yega-Cliente** - Aplicación cliente



Stack Técnico

- **Backend:** PHP 8+ con APIs REST
- **Base de Datos:** MySQL con Prisma ORM
- **Frontend:** JavaScript vanilla + HTML5 + CSS3

- **Visualizaciones:** Chart.js 4.4.0
- **Integración:** GitHub API v3
- **Estilos:** CSS Grid, Flexbox, Animaciones CSS

Instalación Rápida

1. Prerrequisitos

```
# PHP 8.0 o superior
php --version

# MySQL 8.0 o superior
mysql --version

# Composer
composer --version

# Node.js (para Prisma)
node --version
```

2. Clonar y Configurar

```
# Navegar al directorio del proyecto
cd yega-dashboard

# Instalar dependencias PHP
composer install

# Instalar dependencias Node.js
npm install

# Copiar configuración
cp .env.example .env
```

3. Configurar Base de Datos

```
# Crear base de datos
mysql -u root -p
CREATE DATABASE yega_dashboard CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
EXIT;

# Ejecutar migraciones
mysql -u root -p yega_dashboard < database/migration.sql

# Generar cliente Prisma
npx prisma generate
```

4. Configurar Variables de Entorno

Editar archivo `.env`:

```
# Base de datos
DATABASE_URL="mysql://tu_usuario:tu_password@localhost:3306/
yega_dashboard"

# GitHub API (OBLIGATORIO)
GITHUB_TOKEN=ghp_tu_token_personal_aqui

# Configuración app
APP_ENV=production
CACHE_TTL=3600
```

5. Crear Token de GitHub

1. Ve a GitHub → Settings → Developer settings → Personal access tokens
2. Genera nuevo token con permisos:
 - `repo` (acceso completo a repositorios)
 - `read:user` (información de usuario)
3. Copia el token al archivo `.env`

6. Inicializar Datos

```
# Sincronización inicial
php sync.php

# Verificar que los datos se cargaron
mysql -u root -p yega_dashboard -e "SELECT * FROM repositories;"
```

7. Levantar el Servidor

```
# Servidor PHP incorporado
php -S localhost:8000 -t public

# O usar composer
composer run serve
```

8. Acceder al Dashboard

Abrir en navegador: `http://localhost:8000`

Configuración Avanzada

Servidor Web (Apache/Nginx)

Apache

```
<VirtualHost *:80>
    ServerName yega-dashboard.local
    DocumentRoot /ruta/a/yega-dashboard/public

    <Directory /ruta/a/yega-dashboard/public>
        AllowOverride All
        Require all granted

        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^api/(.*)$ api/index.php [QSA,L]
    </Directory>
</VirtualHost>
```

Nginx

```
server {
    listen 80;
    server_name yega-dashboard.local;
    root /ruta/a/yega-dashboard/public;
    index index.html;

    location /api/ {
        try_files <span class="math-inline" style="display:
inline;"><math xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mi>u</mi><mi>r</mi><mi>i</mi></mrow></
math></span>uri/ /api/index.php?$query_string;
    }

    location ~ /\.php$ {
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME <span class="math-inline"
style="display: inline;"><math xmlns="http://www.w3.org/1998/Math/
MathML" display="inline"><mrow><mi>d</mi><mi>o</mi><mi>c</
mi><mi>u</mi><mi>m</mi><mi>e</mi><mi>n</mi><msub><mi>t</mi><mi>r</
mi></msub><mi>o</mi><mi>o</mi><mi>t</mi></mrow></math></
span>fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Cron para Sincronización Automática

```
# Abrir crontab
crontab -e

# Sincronizar cada 5 minutos
*/5 * * * * cd /ruta/a/yega-dashboard && php sync.php >> logs/
cron.log 2>&1

# Sincronizar cada hora (menos agresivo)
0 * * * * cd /ruta/a/yega-dashboard && php sync.php >> logs/
cron.log 2>&1
```



APIs Disponibles

Repositorios

```
GET /api/repositories
# Respuesta: Lista de todos los repositorios con estadísticas
```

Issues

```
GET /api/repository/SebastianVernisMora/Yega-API/issues
# Respuesta: Issues del repositorio especificado
```


Pull Requests

```
GET /api/repository/SebastianVernisMora/Yega-API/pulls  
# Respuesta: PRs del repositorio especificado
```

README

```
GET /api/repository/SebastianVernisMora/Yega-API/readme  
# Respuesta: Contenido del README en markdown
```

Estadísticas

```
GET /api/stats/overview  
# Respuesta: Estadísticas generales de todos los repos
```

Sincronización Manual

```
POST /api/sync  
# Fuerza sincronización inmediata con GitHub
```

Personalización del Tema

Colores Neón

Modificar en `public/css/dashboard.css`:

```
:root {
  --neon-pink: #ff007f;      /* Rosa neón principal */
  --neon-cyan: #00ffff;      /* Cian neón */
  --neon-purple: #8a2be2;    /* Morado neón */
  --neon-green: #32ff32;     /* Verde manzana */
}
```

Agregar Nuevos Repositorios

1. Editar `.env`:

```
YEGA_REPOSITORIES="SebastianVernisMora/Yega-
Ecosistema,SebastianVernisMora/Yega-API,SebastianVernisMora/Tu-
Nuevo-Repo"
```

1. Ejecutar sincronización:

```
php sync.php
```



Estructura del Proyecto

```
yega-dashboard/  
├── src/                                # Código fuente PHP  
│   ├── GitHubAPI.php                 # Cliente de GitHub API  
│   └── Database.php                  # Operaciones de base de datos  
├── public/                            # Frontend público  
│   ├── index.html                    # Página principal  
│   ├── css/dashboard.css             # Estilos neón  
│   ├── js/dashboard.js               # Lógica frontend  
│   └── api/index.php                 # Endpoints REST  
├── database/                          # Migraciones SQL  
├── logs/                              # Logs de aplicación  
├── schema.prisma                     # Schema de Prisma  
├── sync.php                           # Script de sincronización  
├── composer.json                     # Dependencias PHP  
└── package.json                      # Dependencias Node.js
```



Troubleshooting

Error: "Access denied for user"

```
# Verificar credenciales de MySQL  
mysql -u tu_usuario -p  
  
# Otorgar permisos si es necesario  
GRANT ALL PRIVILEGES ON yega_dashboard.* TO  
'tu_usuario'@'localhost';  
FLUSH PRIVILEGES;
```

Error: "API rate limit exceeded"

- Verificar que el token de GitHub sea válido
- Esperar reset del rate limit (1 hora)
- Usar token con mayores límites

Dashboard no carga datos

```
# Verificar logs
tail -f logs/sync.log
tail -f logs/github.log

# Ejecutar sincronización manual
php sync.php
```

Problemas de permisos

```
# Dar permisos a directorio de logs
chmod 755 logs/
chown www-data:www-data logs/
```

Rate Limits de GitHub

- **Con token:** 5,000 requests/hora
- **Sin token:** 60 requests/hora
- **Cache TTL:** 1 hora (configurable)

Actualizaciones

```
# Actualizar dependencias
composer update
npm update

# Regenerar cliente Prisma si hay cambios en schema
npx prisma generate
```

Deployment

Servidor Linux (Ubuntu/CentOS)

1. Instalar stack LAMP:

```
sudo apt update
sudo apt install apache2 mysql-server php8.2 php8.2-mysql php8.2-curl
```

1. Clonar proyecto:

```
cd /var/www/html
sudo git clone [tu-repo] yega-dashboard
sudo chown -R www-data:www-data yega-dashboard
```

1. Configurar Apache:

```
sudo a2enmod rewrite
sudo systemctl restart apache2
```



Logs

- `logs/sync.log` - Sincronización con GitHub
- `logs/github.log` - Requests a GitHub API
- `logs/database.log` - Operaciones de BD
- `logs/cron.log` - Ejecuciones programadas



Soporte

Para reportar issues o solicitar nuevas funcionalidades:

1. Revisar logs de errores
2. Verificar configuración de `.env`
3. Comprobar conectividad a GitHub API
4. Validar permisos de archivos y directorios



Desarrollado para el ecosistema Yega con amor y muchísimo neón 🎆