Website Informe de Gases de Efecto Invernadero

Introducción

Aplicación desarrollada con las siguientes tecnologías:

- PHP 7.1.5
- Zend Framework 3,
- Doctrine 2,
- MySQL 5.7.
- Gestor de dependencias de PHP: Composer
- Gestor de dependencias de frontend: Bower

El proyecto consta con 2 módulos:

Modulo Application

Modulo compuesto de dos paginas HTML, arquitectura MVC.
Ubicado en `./modules/Application`

Tecnologías

HTML5. CSS, PHP, jQuery, D3.js

Estructura básica

- Home del proyecto, punto de entrada al sitio web.
- Página de resultados de los informe de Gases de Efecto Invernadero.

Interfaces del sistema

Interface con el modulo Api, del cual consumen los web services expuestos a través de una serie de endpoints que retornan JSON, con el cual se construyen los gráficos de los reportes de forma dinámica.

Modulo Api

Capa de web services RESTful.
Ubicado en `./modules/Api`

Tecnologías

PHP, MySQL, Doctrine

Consta de una serie de endpoints qconsultados por la pagina de resultados que proveen de datos para los graficos de la pagina de resultados.

Endpoints expuestos

- /informe/distribucion-sectores/:year
- /informe/distribucion-sector/:year/:sector
- /informe/distribucion-gases/:year
- /informe/distribucion-gases-sector/:year

- /informe/evolucion-sectores
- /informe/evolucion-sector/:sector
- /informe/evolucion-sector-subactividad/:sector
- /informe/evolucion-sector-subactividad-categoria/:sector/:subactivity
- /informe/indicador/:indicator

Instalación

Clonar el proyecto

```
$ git clone git@github.com:diegoangel/informe-gei.git
$ cd informe-gei
$ git checkout zf3-version
$ composer install
```

Crear base de datos

Cambiar credenciales de conexión si corresponde y ejecutar:

```
mysql -u root -proot -e "CREATE SCHEMA informe_gei"
```

Configuración

Conexion a base de datos

Los datos de configuracion de la conexion deben colocarlo en el archivo `config/autoload/local.php`.

```
...
return [
    'doctrine' => [
        'connection' => [
            'orm_default' => [
                'driverClass' => PDOMySqlDriver::class,
                'params' => [
                    'host'     => '127.0.0.1',
                    'user'     => 'root',
                    'password' => 'root',
                    'dbname'   => 'informe_gei',
                    'charset'  => 'utf8',
                ]
            ],
        ],
    ],
];
```

Este archivo no existira, por lo cual deben crearlo, copiando, pegando y renombrando el archivo `config/autoload/local.php.dist`.

```
cp config/autoload/local.php.dist config/autoload/local.php
```

Además, este archivo es ignorado en el repositorio de control de versión y por lo tanto las credenciales de conexión nuncan seran compartidas por accidente y permanecen seguras.

Doctrine Migrations

Se utiliza la funcionalidad del proyecto Doctrine Migrations para gestionar los cambios y versionado de la base de datos.
Para conocer mas acerca de las posibilidades o funcionamiento de esta herramienta sugerimos visitar el enlace a la documentación del proyecto.

Una vez configurada la base de datos proceda a la carga inicial de datos.

Ejecutar el siguiente comando para iniciar la misma:

```
./vendor/bin/doctrine-module migrations:migrate
```

A continuacion se le informara que esta por ejecutar el comando y su posible criticidad y debera confirmar el mismo.

```
Loading configuration from the integration code of your framework (setter).

                    Doctrine Database Migrations


WARNING! You are about to execute a database migration that could result in schema changes and da
ta lost. Are you sure you wish to continue? (y/n)
```

Presione la letra **y**

Los parametros de configuración de las Migration de Doctrine se encuentran en el archivo `config/autoload/global.php`

```
...
return [
    'doctrine' => [
        // migrations configuration
        'migrations_configuration' => [
            'orm_default' => [
                'directory' => 'data/Migrations',
                'name'      => 'Doctrine Database Migrations',
                'namespace' => 'Migrations',
                'table'     => 'migrations',
            ],
        ],
    ],
];
```

Iniciar servidor de desarrollo

Iniciar el servidor de consola en el puerto 8080 y navegarlo en http://localhost:8080/ para comprobar el normal funcionamiento del sitio web.

Nota: El servidor embebido de PHP *es solo para desarrollo*, en producción utilizar Apache, Nginx u otro que conforme sus requerimientos.

```
$ php -S 0.0.0.0:8080 -t public/ public/index.php
```

```
# O utilizar el alias definido en la sección scripts de composer.json:
$ composer serve
```

Modo desarrollo para entorno local

El proyecto viene con un modo de desarrollo por defecto, útil para declara configuracion que solo se ejecutara en el entorno local del desarrollador a fin de, por ejemplo, poder establecer una conexión a una base de datos de pruebas entre otras cosas.
Provee tres alias para habilitar, deshabilitar y consultar el estado:

```
$ composer development-enable  # habilita el modo desarrollo
$ composer development-disable # deshabilita el modo desarrollo
$ composer development-status  # informa sobre si esta o no habilitado el modo desarrollo
```

Test unitarios

Se crearon dos suites de test unitarios utilizando PHPUnit.

- Api test suite `./module/Api/test`
- Application test suite `./module/Application/test`

Ejecución de los test

Ejecutar el siguiente comando en consola.

```
$ ./vendor/bin/phpunit  -v --debug --testsuite Api
$ ./vendor/bin/phpunit  -v --debug --testsuite Application
```

Pueden ejecutar ambas suites con el comando:

```
$ ./vendor/bin/phpunit  -v --debug
```

Test coverage

Para generar el reporte de test coverage ejecutar el siguiente comando en consola y luego abrir el archivo `./data/coverage/index.html` en el navegador.

```
$ ./vendor/bin/phpunit -v --debug --coverage-html data/coverage
```

Si necesitan agregar modificaciones locales en la configuracion de PHPUnit, copiar `phpunit.xml.dist` a `phpunit.xml` y editar el nuevo archivo; el ultimo tiene precedencia sobre el primero cuando se ejecutan los test y es ignorado por el sistema de control de versiones.
Si se quiere editar permanentemente la configuracion editar el archivo `phpunit.xml.dist`.

Integración Continua (CI)

Se utilizó Scrutinizer CI para ejecutar:

- Test de Integración.
- Test Unitarios.

- Test Coverage.
- Static Code Analisis.
- Chequeo de vulnerabilidades.
- Detección automatica de Malas Prácticas y Bugs.
- Complejidad Ciclomatica.

Pueden encontrar el archivo de configuración `scrutinizer.yml`, que utilizamos en el servidor de integración continua de Scrutinizer.

La evaluación de la calidad del código, la cobertura de tests y el estado del ultimo build se pueden visualizar a traves de los badges que se encuentran debajo del título del documento.

NOTA: Se puede acceder a información adicional tales como imagenes, documentos técnicos y capturas de pantalla con informacion de reportes de del proyecto en la carpeta ./doc

Configuración servidor web

Configuracion virtual host en Apache

Ejemplo funcional de configuracion en Apache.

No se contempla en este ejemplo la declaracion de niveles de logueo, ubicacion de archivos de logs o reglas especiales de rewrite por ejemplo.

```
<VirtualHost *:80>
    ServerName informe-gei.local
    DocumentRoot /var/www/html/informe-gei/public

    <Directory /var/www/html/informe-gei/public>
        DirectoryIndex index.php
        AllowOverride All
        Order allow,deny
        Allow from all
        <IfModule mod_authz_core.c>
            Require all granted
        </IfModule>
    </Directory>
</VirtualHost>
```