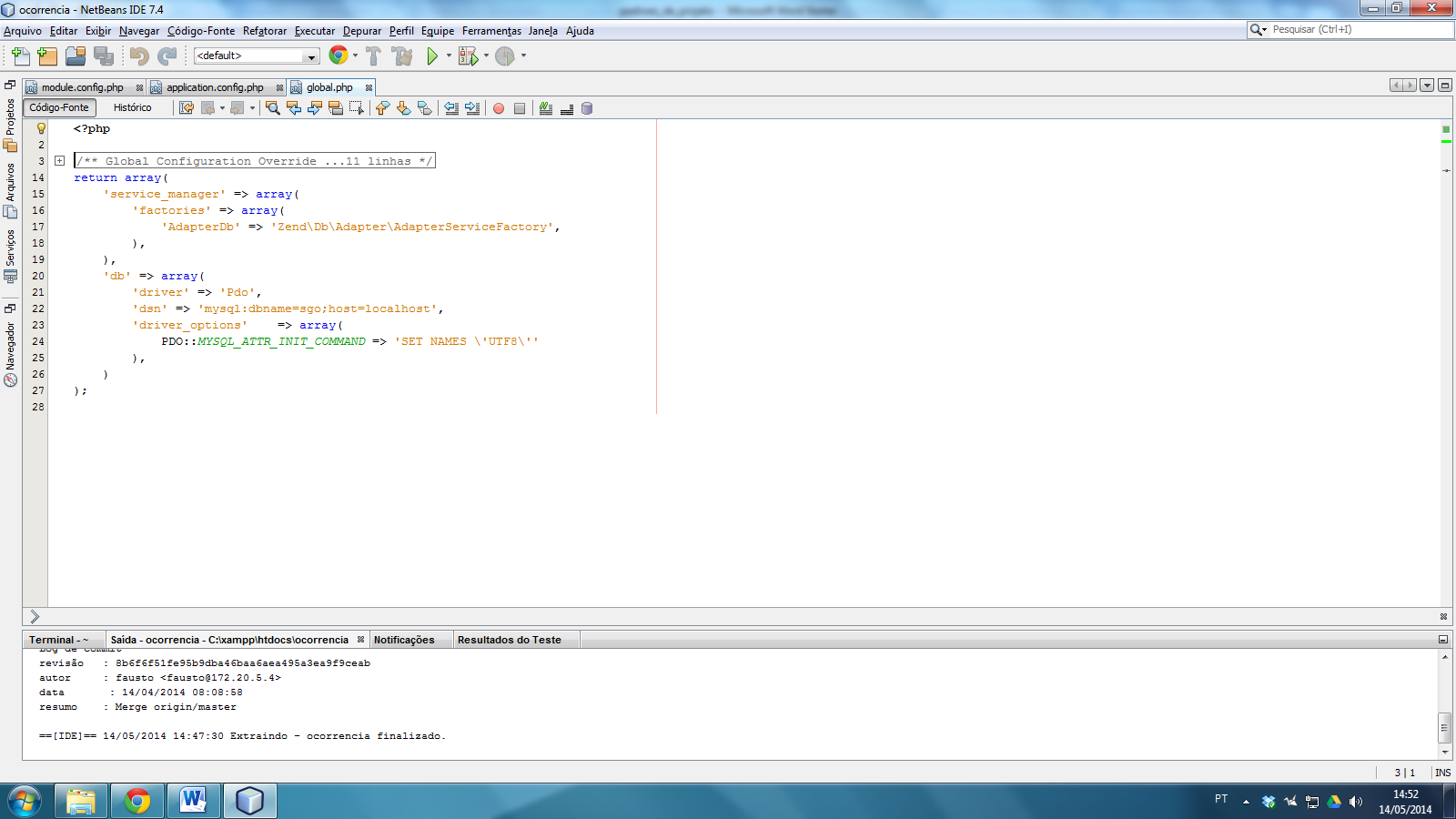
**Injeção de Dependência**

Injeção de dependência significa dizer que os componentes de uma classe não necessitam procurar por suas dependências num contexto global ou usando localizadores de serviços. Ao invés disso, suas dependências são simplesmente injetadas na hora da criação e seu Criador é o responsável por provê-las.

O Zend Framework 2 possui um *service manager*, ou gerenciador de serviços, que permite acesso aos serviços registrados na aplicação. Muitas vezes é necessário ter acesso a uma classe, porém ela possui diversas dependências, logo, para evitar ter que instanciar diversas classes e tratar das dependências de forma manual, é possível fazer todo esse tratamento durante o processo de registro.

Dessa forma quando for preciso recuperar um serviço em um controller, por exemplo, basta invocar o método getServiceLocator()->get(‘Nome\Do\Servico’);

No sistema, esse padrão pode ser observado no arquivo (ocorrência/config/autoload/global.php)



**DAO (Data Access Object)**

Os objetos DAO formam uma camada de lógica de mapeamento misturada com lógica de persistência e lógica de negócio.

Existem algumas características desejáveis em um implementação DAO:

* Todo o acesso aos dados deve ser feito através de classes DAO de forma a se ter o encapsulamento
* Cada instância é responsável por um objeto de domínio
* Deve ser responsável pelas operações CRUD no domínio
* Não deve ser responsável por transações, sessões ou conexões que devem ser tratadas fora da DAO

No sistema, as classes para mapear os objetos de domínio que irão ser persistida (as classes DAO) foram criadas seguindo a seguinte nomenclatura (NomeTable), por exemplo, MunicipioTable:

