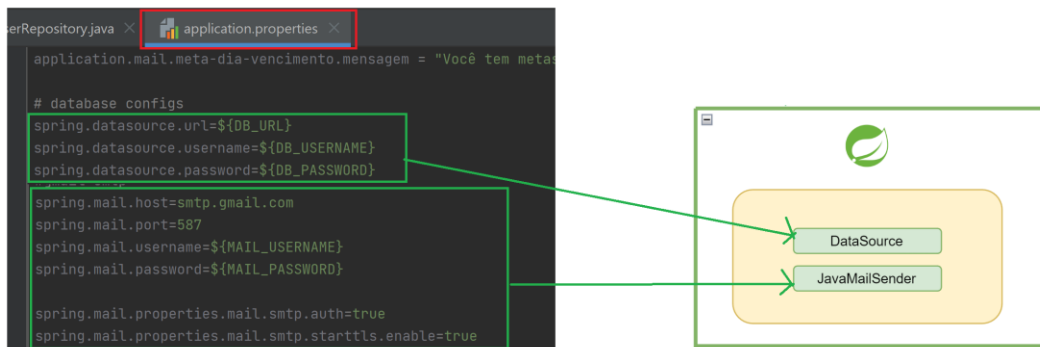




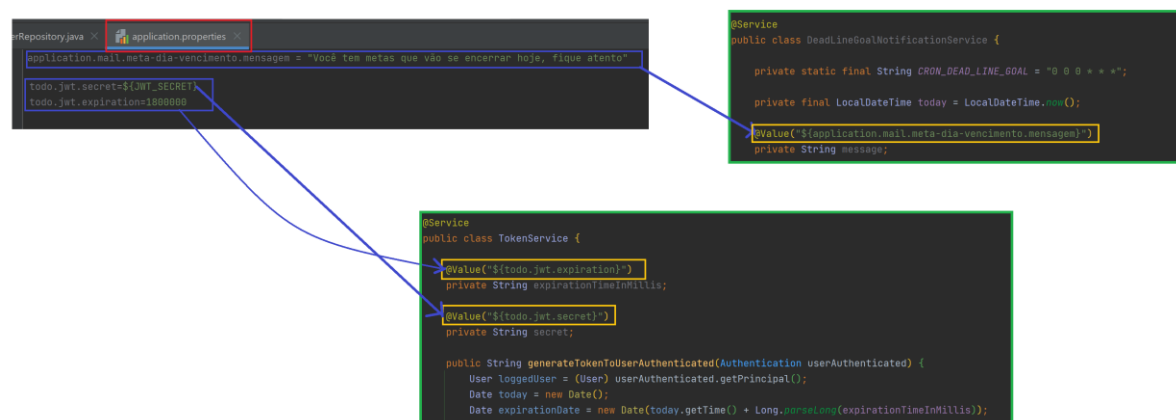
Application properties

O arquivo de **application properties** de uma aplicação **Spring Boot** é como se fosse um **core de configurações relevantes**. Por exemplo, você pode **definir o valor de Beans de diversos tipos**, **Beans de conexão**, **Beans de envio de email** e etc. Esses valores nada mais são do que atributos de **Beans** gerenciados pelo Spring:

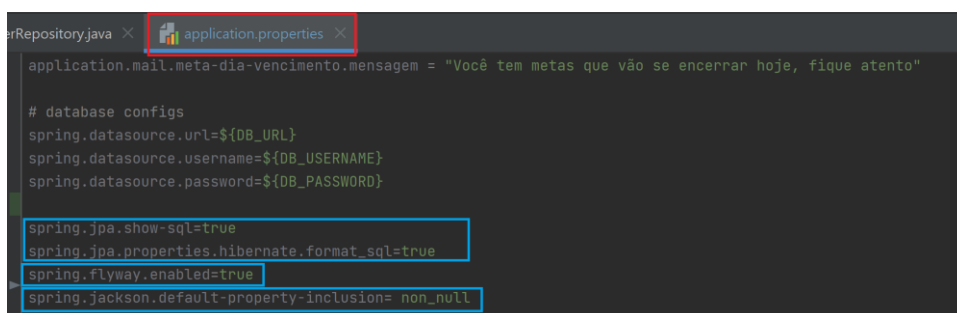


Além de **definir valores de Beans específicos** o properties pode **conter valores relevantes para a segurança da aplicação**, evitando que **informações sensíveis permaneçam no código**, os próprios valores de senha dos Beans acima são um exemplo disso.

Porém, também podemos ter **valores que não são associados diretamente a um Bean**, mas que são **injetados no código através de anotações**, como por exemplo, **informações do JWT** ou até mesmo **um valor de uma mensagem de email**:



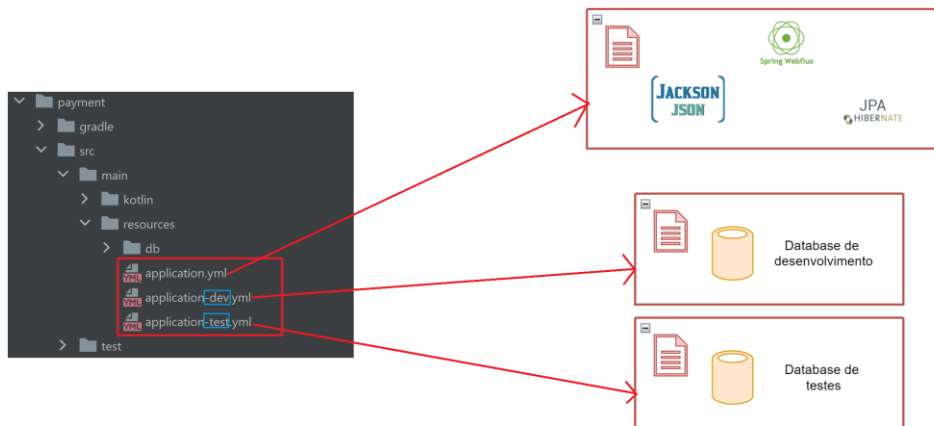
O **properties** é tão poderoso associado ao **Spring Boot** que juntos podem manipular até mesmo **propriedades de bibliotecas terceiras**, como **definir comportamentos do flyway**, do **jpa** e do **Jackson**:



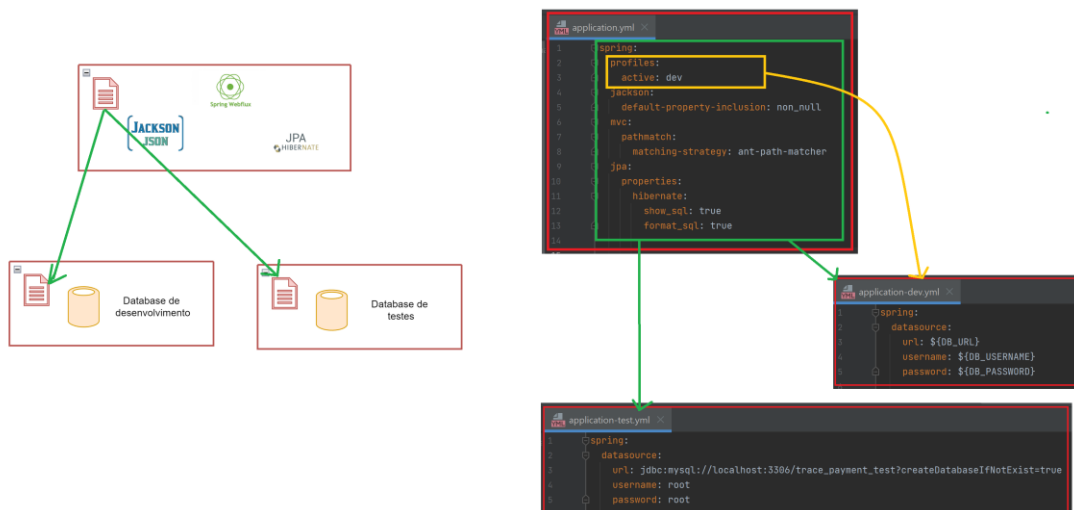
Application Properties e profiles

Sabendo que todo o core de configuração de um projeto pode ser descrito no **application properties** é comum que haja vários arquivos de properties que separam **responsabilidades de acordo com o ambiente**.

Esses **ambientes também podem ser chamados de profile**, o **application que não tiver nenhum tipo de prefixo vai ser reconhecido como o padrão**, os demais podem ter **prefixos associados que indicam que deixem claro qual ambiente vai utilizar aquelas propriedades de aplicação** (é uma convenção, mas poderiam ter outro nome):



O **application properties padrão** pode abrigar **configurações gerais que serão herdadas pelos outros arquivos de application**, o arquivo padrão também **pode definir qual application estará ativo quando a aplicação for executada**:



Existem outras **maneiras de executar a aplicação ou partes dela com algum perfil específico**, por exemplo, uma **anotação** na classe de testes **que indique qual perfil ela deve usar** ao ser executada:

```
@ActiveProfiles("test")
@AutoConfigureMockMvc
@ExtendWith(SpringExtension.class)
@SpringBootTest
@WithMockUser(username="admin",roles={"USER","ADMIN"})
class GoalControllerTest {
```