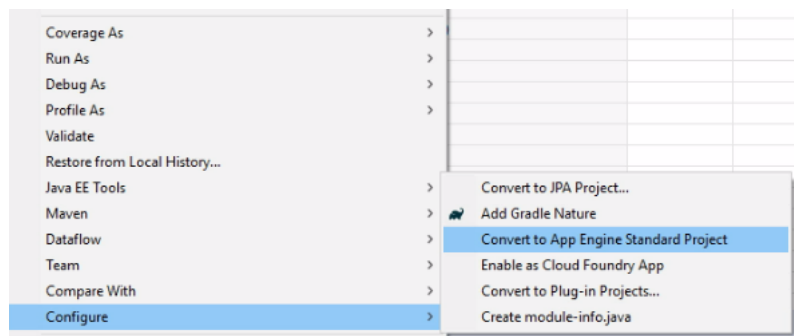
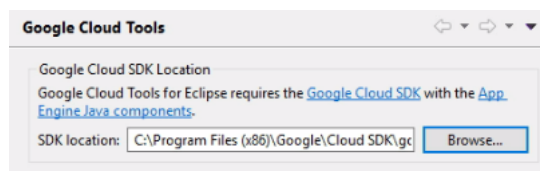


Mãos à obra: Convertendo projeto

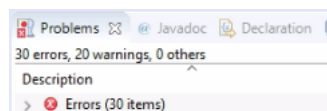
Para levarmos nosso projeto para o Google Cloud através do App Engine, o primeiro passo é converter o projeto para que ele seja um projeto do App Engine, para isso, clique com o botão direito do mouse no projeto da Casa do Código e selecione a opção **Configure -> Convert to App Engine Standard Project**.



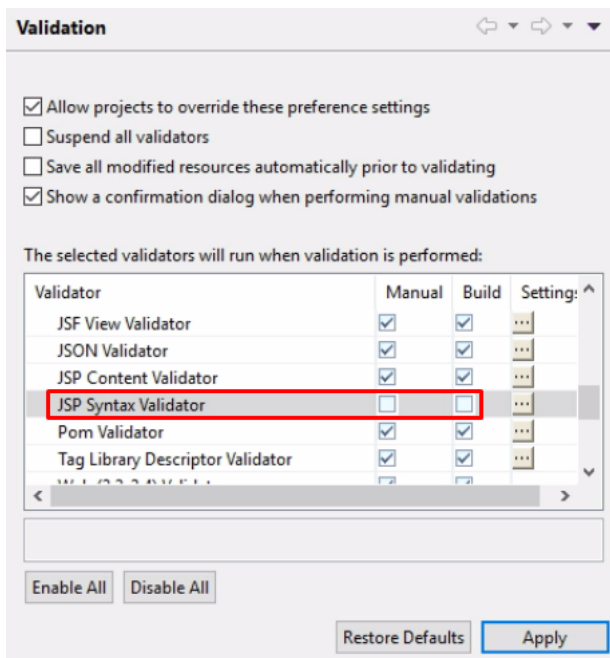
Caso o Eclipse não encontre o local onde o Cloud SDK foi salvo, uma nova tela abrirá para especificarmos esse caminho completo.



Feito isso, nosso projeto será agora do tipo App Engine, e podemos levá-lo para o Google Cloud. Porém, para que essa integração seja possível, não podemos ter nenhum tipo de erro no nosso projeto. Se formos até a aba **Problems** no Eclipse, vamos ver que alguns erros foram alarmados, tais erros seriam na verdade que o Eclipse não conseguiu reconhecer a sintaxe de algumas *expression languages* usadas na JSP, porém não são erros no projeto.



Para corrigir esse cenário e fazer a implementação no Google Cloud, vamos configurar o Eclipse para não fazer a verificação de sintaxe das JSPs em nosso projeto. Para isso, clique com o botão direito do mouse no projeto e selecione a opção **Properties** e pesquise na nova janela que abrir a parte de validações, **Validation**. Na sequência, clique em **Configure Workspace Settings** e desmarque a opção referente a validação da sintaxe das JSPs.



The Validation dialog box in Google Cloud IDE. It contains several checkboxes for validation preferences and a table of validators.

Validation

☒ Allow projects to override these preference settings
☐ Suspend all validators
☐ Save all modified resources automatically prior to validating
☒ Show a confirmation dialog when performing manual validations

The selected validators will run when validation is performed:

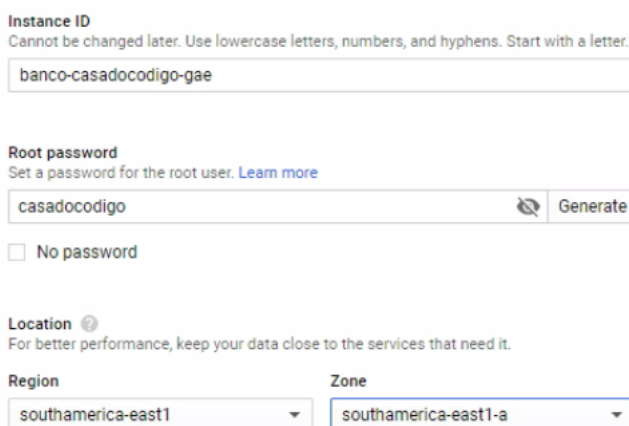
Validator	Manual	Build	Setting: ^
JSF View Validator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	...
JSON Validator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	...
JSP Content Validator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	...
JSP Syntax Validator	<input type="checkbox"/>	<input type="checkbox"/>	...
Pom Validator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	...
Tag Library Descriptor Validator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	...

Buttons: Enable All, Disable All, Restore Defaults, Apply

Na sequência, clique em **Apply** para que as mudanças surtam efeito. Nesse momento, não deveremos ter mais nenhum tipo de erro no projeto.

Agora que já temos nosso projeto configurado, não podemos nos esquecer que essa aplicação vai precisar se comunicar com o banco de dados e salvar as imagens dos livros no Bucket do Cloud Storage. Vamos voltar para o painel de console do Google Cloud e configurar o banco de dados, certifique-se que você está agora no projeto **casadocodigo-gae** e vá até a parte do Cloud SQL.

No Cloud SQL faça as mesmas configurações que fizemos nas etapas anteriores, escolha o banco MySQL e a máquina de segunda geração para alocar esse banco de dados. Posteriormente, escolha o nome para o banco, por exemplo **banco-casadocodigo-gae**, coloque uma senha para que possamos acessar essa instância, por exemplo **casadocodigo** e especifique a região e zona onde essa instância deverá ser configurada, **southamerica-east1** e **southamerica-east1-a**, respectivamente.




Instance ID
Cannot be changed later. Use lowercase letters, numbers, and hyphens. Start with a letter.
banco-casadocodigo-gae

Root password
Set a password for the root user. [Learn more](#)
casadocodigo
☐ No password



Location
For better performance, keep your data close to the services that need it.

Region: southamerica-east1
Zone: southamerica-east1-a


Posteriormente, selecione as redes autorizadas a realizar conexão com essa instância para que possamos criar o banco **casadocodigo** que será utilizado por nossa aplicação.

4 Authorize networks 

Authorized networks
Add IPv4 addresses below to authorize networks to connect to your instance. Networks will only be authorized via these addresses.

New network  

Name (Optional)

Network
Use CIDR notation. 

Feito isso, clique no botão **Create** para criar a instância com o banco de dados, espere alguns minutos até que a instância tenha sido completamente inicializada. Na sequência, copie o endereço IP público e altere o método **dataSource** na classe **JPAConfiguration**, para realizar a comunicação com esse novo banco.

@Bean

```
public DataSource dataSource() {  
    DriverManagerDataSource dataSource = new DriverManagerDataSource();  
    dataSource.setUsername("root");  
    dataSource.setPassword("casadocodigo");  
    dataSource.setUrl("jdbc:mysql://[Endereço IP público instância]:3306/casadocodigo");  
    dataSource.setDriverClassName("com.mysql.jdbc.Driver");  
    return dataSource;  
}
```

Agora alteraremos a classe **FileSaver** para salvar as imagens no Bucket configurado pelo App Engine. Para isso, vá até a aba Cloud Storage e localize o Bucket configurado pelo App Engine.

```
BlobInfo blobInfo = storage.create(  
    BlobInfo.newBuilder("[Bucket configurado pelo App Engine]",  
        file.getOriginalFilename()).build(),  
    file.getBytes(),  
    BlobTargetOption.predefinedAcl(PredefinedAcl.PUBLIC_READ));  
return blobInfo.getMediaLink();
```

Na sequência, vamos fazer um teste para confirmar se nossa aplicação continua funcionando localmente para que depois façamos o envio da aplicação para a plataforma do Google Cloud.