Passo	Teste/Verificação	Tempo de ref. máx.	OK	NOK
Passo 1	Crie um projeto vazio no seu IDE para o projeto importando de seguida para o seu projeto (vazio) o projeto da sua implementação (a partir do repositório individual *(GitHub)*, onde estão os *sources* de referência correspondentes à implementação submetida na data de entrega do trabalho.	30 seg	X	
Passo 2	Após ter importado o seu projeto no Eclipse (ou no seu IDE que preferir), deverá mostrar ser capaz de preparar tudo para fazer a compilação e deployment local, demonstrado que o componente servidor fica a executar localmente (local host) e que pode já ser usado pela aplicação cliente (browser) que lhe consegue aceder. Após este passo, será suposto que o seu projeto está pronto para deployment remoto no ambiente CLOUD - para isso pode imediatamente mostrar que é possível executar uma qualquer operação (por exemplo um login como super-user) no servidor no deployment local.	30 seg		X
Passo 3	Faça agora *deployment* do projeto para ambiente Cloud (Google App Engine), mostre que fica pronto para poder ser utilizado a partir do servidor remoto (Cloud) para poder prosseguir nos próximos passos da demonstração/verificação sobre o servidor remoto (Google Cloud). Notar que a sua aplicação *deployed* não pode ter inicialmente nenhum utilizador registado (a não ser o bootstrap user com role SU de acordo com os requisitos do enunciado do projeto). Mostre com a evidência que achar oportuna que o servidor remoto Cloud está pronto e funcional, por exemplo mostrando que é possível o utilizador superuser usar o browser, fazer login e entrar numa sessão autenticada do servidor deployed e executando na nuvem.	30 seg	X	
Passo 4	Mostre neste passo que é possível três utilizadores diferentes (default roles: USER) conseguirem criar contas (mostrando como) no seu sistema. Os utilizadores devem ter como usernames o vosso próprio nome e sobrenome e número e um sufixo U, GBO, GA, por ex; (exemplo ruimanuel56451U, ruimanuel5645GBO, ruimanuel5645GA. Mostre que é impossível realizarem LOGIN pois as contas estão INATIVAS. Após a alteração do estado para ATIVO por parte do SU, mostre que após a criação das contas os utilizadores vão conseguir fazer LOGIN e LOGOUT em sessões na sua aplicação. Mostre esses utilizadores conseguem fazer login e logout e no caso de um deles aproveite para mostrar os atributos registados na criação da conta	1m30 a 2min	X	
Passo 5	Com os anteriores utilizadores logged-out, utilize agora uma sessão do user SU para promover os utilizadores criados aos seguintes ROLES: ruimanuel5645GBO deverá ser promovido a GBO ruimanuel5645GA deverá ser promovido a GA O utilizador ruimanuel5645U permanecerá com o role USER Note que isso neste passo só deve poder ser feito pelo utilizador bootstrap "superuser", uma vez que é o único role com autorização para proceder a mudanças de role	30 seg	X	

Passo 6	Volte a fazer login com o utilizador que ficou com role USER. Mostre que o utilizador pode mudar a sua password e modificar/adicionar um ou mais atributos do registo e quais os atributos que suporta e que podem ser modificados. Faça depois durante 2 a 3 minutos uma demo em que mostre todas as operações que este utilizador (no seu role USER) pode fazer na sua aplicação quando está em sessão, mostrando assim as funcionalidades que conseguiu implementar e que correspondem às operações do enunciado para o role USER. No final faça logout e mostre que o token ou cookie (da sessão) não vai poder ser reutilizado e que qualquer operação desse utilizador vai requerer novo LOGIN. Faça LOGIN e deixe este utilizador em sessão (convém que o token ou cookie tenha uma validade relativamente grande da ordem de 15-20 minutos para não expirar durante os próximos passos do guião).	~ 2 min	X	
Passo 7	Repita agora o passo 6 para o utilizador com role GBO. Neste caso mostrará as funcionalidades que este pode realizar, demonstrando as operações suportadas e em que fique claro as operações que um utilizador GBO pode realizar mas o utilizador USER não pode (mostrando que na sessão em que este está, a tentativa de fazer uma operação apenas acessível ao GBO vai dar erro por falta de permissão.	~2 min	X	
Passo 8	Neste passo deve saber explicar como está a emitir e gerir os *tokens* ou *cookies* e como estes possuem validade controlada na sessão (persistência, estado na "base de dados" e controlo de validação/invalidação) garantindo que são invalidados corretamente numa operação de logout e mostrando qual o retorno e o controlo na operação de *logout. Para tal vai fazer login com o utilizador com o role GA e mostrar a operação que permite mostrar o conteúdo do token ou cookie de sessão, tal como foi emitido no LOGIN.* A ideia é mostrar que não vai ser possível após um logout ficar com dados no ecrã da sessão anterior (por exemplo se tiver clientes em Browser/Javascipt) ou que nesse caso um mero refresh força a ter que fazer novo login (na página login). Note que se espera que após um logout, o browser volte sempre para a página de LOGIN. Mostre para confirmar que o simples refrescamento, cópia de URLs, ou repetição dos pedidos (URLs+Gets ou Posts), ou "andar para trás" no browser não permite obter a sessão "passada" desse utilizador e não vai permitir realizar qualquer operação ... e que (preferivelmente) esta tentativa forçará à redirecção do utilizador para a página de LOGIN. No caso de quer mostrar isso com o Postman, mostre que se usar um token ou cookie anterior (após um logout) não consegue executar operações com o utilizador em questão.	~2 min	X	
Passo 9	Mostre que o utilizador GBO não tem como remover utilizadores de role GA, mas que um utilizador com role GA consegue remover a conta do utilizador GBO. Remova o utilizador GBO e mostre depois que esse utilizador já não conseguirá fazer login. Será interessante mostrar que o GBO fica sem acesso mesmo que esteja numa sessão autenticada com um token ou cookie válido no momento em que é removido. Para o efeito, quando tentar fazer qualquer operação será forçosamente fechada a sessão atual. Isto é, encontrando-se o GBO numa sessão (válida), mal o utilizador GA remova ou desabilite a conta do GBO, a sessão deste deve terminar deixando o GBO de poder usar o sistema.	~2 min		X

Passo 10	Mostre que o utilizador USER na sessão, não pode remover o utilizador GA. Mostre que existem operações que um utilizador com perfil USER pode fazer mas que não aparecem na sessão de um utilizador GBO (exemplo: operação op4, conforme referido no enunciado)	1 min	X	
Passo 11	Neste passo apenas terá que explicar (mostrando no código) como está concretizado o modelo de controlo de acesso para os utilizadores com os diferentes roles e como está a gerir as permissões de cada um desses roles, ficando claro que cumpriu os requisitos do enunciado. Mostre também que os utilizadores criados e apagados bem como os tokens ou cookies de sessão estão convenientemente geridos com persistência (base de dados) e se/como/porquê o estado da aplicação será preservado, sobrevivendo à necessidade de reboot da aplicação	~2 min	X	
Passo 12	Aproveite agora para mostrar e avaliar criticamente o "peso" (latência e desempenho) das suas operações escolhendo uma ou mais operações exemplificativas chamadas na sessão de um utilizador. Neste caso pode usar para o efeito a ferramenta postman, a consola de desenvolvimento do seu browser (web inspection tools) ou ambas as ferramentas. Deve estar preparado neste passo para demonstrar qual a latência de 4 ou 5 operações (invocações REST) à sua escolha como referencial de desempenho da sua aplicação. Identifique, da latência observada, quanto se deve à rede (acesso Internet) e quanto se deve à execução das operações propriamente ditas no servidor.	1 min	X	
Passo 13	Apresente e demonstre qualquer funcionalidade (operação, feature, solução que criou e que implementou) que queira destacar como valorizável e que tenha feito como extra, em relação ao enunciado base. Se não tem operações extra basta escrever X na coluna NOK deste passo.	2 min		X