Requisitos e Composição da Nota

Pontuação	Requisito
Back-end e Banco de Dados (4,0 pts)	
1,0	A API deverá ser implementada em **Python e com Flask** com pelo menos 3 rotas (por exemplo, *"/cadastrar_usuario"*, *"/buscar_usuario"* e *"/deletar_usuario"*), sendo que pelo menos uma delas deve implementar o método POST (por exemplo, na rota de cadastro).
1,0	Fazer o uso de um banco de dados **SQLite, PostgreSQL** ou **MySQL** com pelo menos uma tabela (por exemplo, tabela de usuários cadastrados). **Observação**: caso seja utilizado PostgreSQL ou MySQL, deve ser entregue um script Python ou SQL para criação, configuração e carga inicial de dados. O script deve estar bem localizado no projeto e todas as orientações para execução devem estar claras no arquivo README.md
1,0	Qualidade da Documentação da API em Swagger.
1,0	Criatividade e Inovação
Front-end (4,0 pts)	
1,5	Desenvolvimento de uma **SPA (Single Page Application)** utilizando HTML, CSS e JavaScript
1,0	Criatividade e Interativa do front proposto
0,5	Que na tela seja apresentado itens em listas (por exemplo, lista de usuários ou livros)
1,0	Que, ao longo do código, seja feita a chamada a todas as rotas implementadas pela API.
Organização dos Códigos (2,0 pts)	
0,75	Devem ser criados **dois projetos separados**: um para a API e outro para o front-end. Cada projeto deve estar em um repositório git próprio.
0,75	Em ambos os repositórios deve existir um arquivo README.md com apresentação da aplicação e com todas as orientações para configuração e execução do código.
0,5	Qualidade da organização do código.

Atenção! *Caso sejam utilizados como base os exemplos apresentados em aula, serão penalizadas as entregas que não apresentarem pelo menos 50% de código novo.*