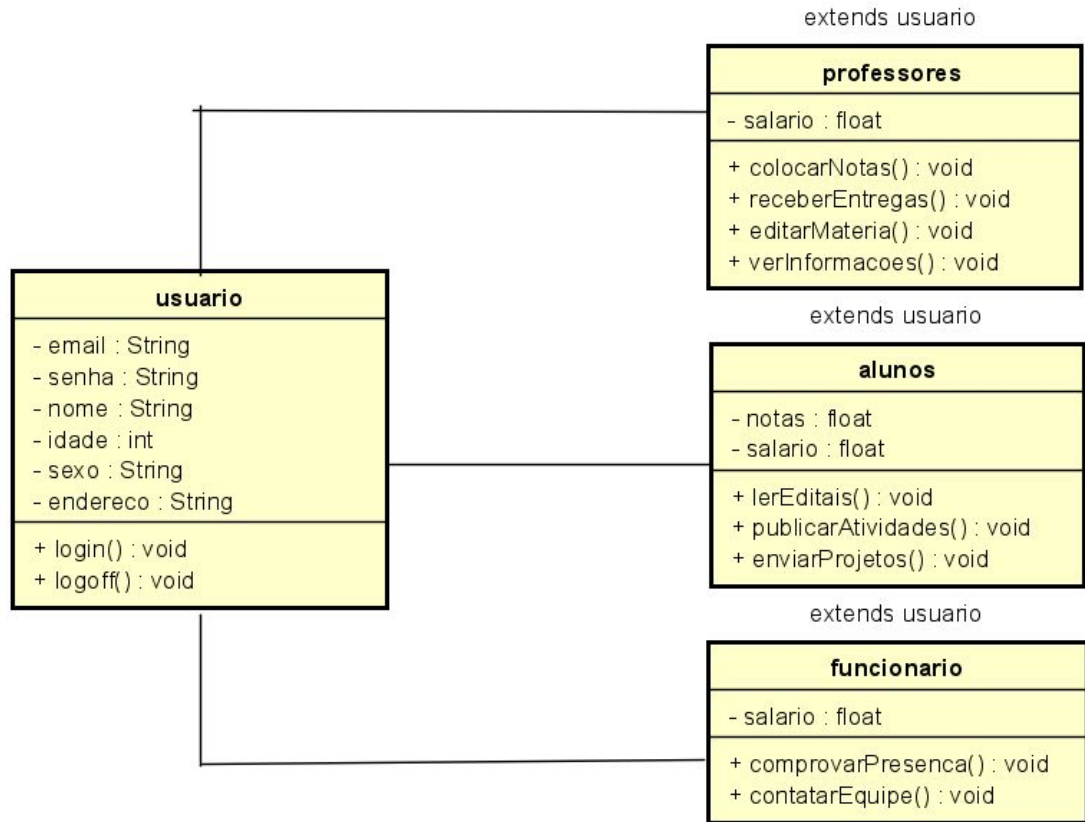
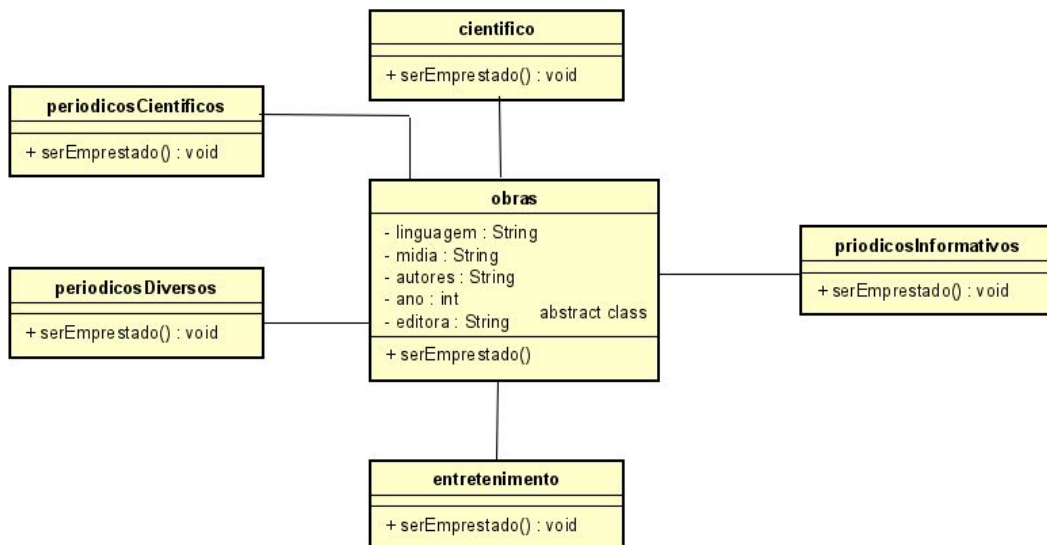


## Capítulo 1 - Introdução a Orientação a Objetos

### 1 - Usuário (professores, alunos e funcionário)



Obras:



OBS: poderia ser usado `@Override`

2 -

Complexidade no aprendizado para desenvolvedores de linguagens estruturadas

- Maior uso de memória (heap), por exemplo para aplicações móveis em Java ME.
- Maior esforço na modelagem de um sistema OO do que estruturado (porém menor esforço de codificação, sendo uma vantagem)
- Funcionalidades limitadas por interface, quando estas estão incompletas (problemas na modelagem)
- Dependência de funcionalidades já implementadas em superclasses no caso da herança, implementações espalhadas em classes diferentes

fonte: <https://wpjr2.wordpress.com/2008/04/23/orientacao-por-objetos-vantagens-e-desvantagens/>

Coesão:

Coesão se trata de dar ações/responsabilidades úteis e lógicas para suas classes e métodos, deixando assim a arquitetura mais coesa possível, um programa que no qual não se foi aplicada a coesão ficaram com suas classes e métodos poucos relacionados.

Acoplamento:

Se baseia em “acoplar” classes para que elas fiquem relacionadas por algum motivo, no entanto, isso acaba dificultando a manutenção do sistema, pois, se há algo de errado ou algo que precisa ser mudado por quaisquer sejam os motivos, o desenvolvedor correrá o risco de mexer em uma classe que pode estar acoplada e acabar, deixando tudo pior.