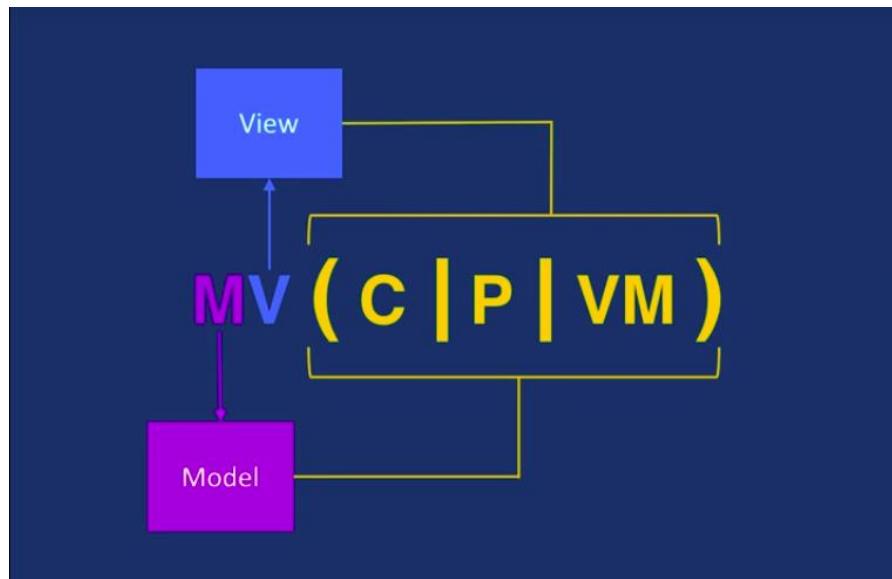


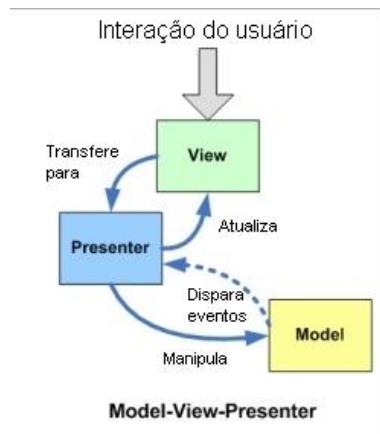
MVP



Em aplicações que possuem a presença de um cliente, é necessário que a arquitetura do software permita que o cliente possa interagir com os dados e alterá-los. Dessa forma, é ideal que haja uma divisão da aplicação a partir da arquitetura, em que os dados estejam separados da sua manipulação e da apresentação ao cliente, facilitando a sua escalabilidade.

Diante disso, surge o conceito dos padrões de projetos: MVC, MVP e MVVM, que aplicam a divisão da aplicação em subsistemas em suas arquiteturas buscando obter um fluxo de dados claro, manutenibilidade e um código mais legível.

Model View Presenter (MVP)



A arquitetura MVP é uma arquitetura baseada em componentes criados a partir de uma tríade de objetos: *models*, *views* e *presenters*. Este padrão é semelhante ao MVC, em que o

Model não sofre alterações muito relevantes, a *View* delega eventos ao *Presenter* e atualiza o estado de exibição, e o *Presenter* é essencialmente o *Controller* do MVC, exceto que o seu relacionamento com a *View* é feito através de uma interface, não gerenciando o tráfego de solicitações recebidas. Ou seja, em linhas gerais o *Presenter* faz a ligação entre a *View* e o *Model*, e possui um papel de mediador entre eles. Ele é encarregado de atualizar a *View* quando o *Model* é alterado e de sincronizar o *Model* em relação a *View*.

Nesta arquitetura, existe uma clareza e uma simplicidade maior nas ações do *Presenter*, apenas encarregando a *View* o que exibir, além de uma facilidade maior de testar o *Presenter* e a sua lógica, devido ao seu maior desacoplamento aos outros subsistemas.

Porém, como nem tudo são flores, o padrão também sofre com algumas fraquezas, com problemas na manutenibilidade do código com o tempo e alta imprevisibilidade, assim como o MVC, acarretando assim em *Presenters* gigantescos difíceis de separar.

O MVP é muito utilizado na comunidade Android, além de ser utilizado pela Microsoft em aplicações em .NET como Windows Forms e ASP.NET, além de poder ser utilizado em frameworks baseados em UI (User Interface) como o Angular.

Referências

O padrão MVP, DevMedia, em <https://www.devmedia.com.br/o-padroao-mvp-model-view-presenter/3043>, acesso em 06/06/2024.

Model-view-presenter, WikiPédia, em <https://pt.wikipedia.org/wiki/Model-view-presenter>, acesso em 06/06/2024.

Model-View-Presenter (MVP), AppMaster, em <https://appmaster.io/pt/glossary/model-view-presenter-mvp-pt>, acesso em 06/06/2024.